

```

EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFFFFFFFFFFFFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFFFFFFFFFFFFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEEEEEEEEEEEEE DDD DDD FFFFFFFF
EEEEEEEEEEEEEE DDD DDD FFFFFFFF
EEEEEEEEEEEEEE DDD DDD FFFFFFFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEE DDD DDD FFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFF
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDD FFF

```

[illegible]



```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```

[illegible]



[ IDENT ('V04-000'),

( ++

```
*****
*
*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*   ALL RIGHTS RESERVED.
*
*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*   TRANSFERRED.
*
*   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*   CORPORATION.
*
*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

FACILITY: VAX/VMS EDF (EDIT/FDL) UTILITY

ABSTRACT: This facility is used to create, modify, and optimize  
FDL specification files.

ENVIRONMENT: NATIVE/USER MODE

AUTHOR: Ken F. Henderson Jr.

CREATION DATE: 27-Mar-1981

MODIFIED BY:

V03-010	RRB0009	Rowland R. Bradley	22 Jan 1984
		Enhancement for display of # buckets in index, # of Pages to cache index, and average # key exams.	
V03-009	KFH0009	Ken Henderson	10 Sep 1983
		Support for named UICs.	
V03-008	KFH0008	Ken Henderson	8 Aug 1983
		Changes for seperate compilation.	
V03-007	KFH0007	Ken Henderson	26 Apr 1983
		Added FOUND_0, PRIMARY_INDEX_BUCKETS. Added a bunch of initial values; to reduce the code in INIT_EDITOR. Added SEC_ATTR, NUMBER_KEYS_SET.	



EDFVAR  
V04-000

Source Listing

J 6  
16-Sep-1984 00:42:36  
5-Sep-1984 13:39:37

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFVAR.PAS;1 (1) Page 2

V03-006	KFH0006	Ken Henderson	14 Apr 1983
	Added SET FUNCTION, GRANULARITY, PROMPTING, RESPONSES, JOURNAL_FILE, JOURNAL_ENABLED, support for SEGMENTED keys.		
V03-005	KFH0005	Ken Henderson	31 Jan 1983
	Changed reference of FDL\$TYPE to FDL3\$TYPE.		
V03-004	KFH0004	Ken Henderson	20 Jan 1983
	Removed DASH, and added DEPTHPOINT_LEFT,_MID,_RIGHT.		
V03-003	KFH0003	Ken Henderson	8 Sept 1982
	Consolidated many main variables into the xDATA arrays.		
V03-002	KFH0002	Ken Henderson	2 April 1982
	Added EDF\$GL_SECNUM.		
V03-001	KFH0001	Ken Henderson	23-Mar-1982
	Took out EDITEDL.		

-- )



```
0085 ENVIRONMENT ('LIB$EDFVAR'),
0086
0087 INHERIT (
0088
0089 'SYSS$LIBRARY:STARLET',
0090 'SHR$LIB$FDL$PARDEF',
0091 'LIB$EDFCONST',
0092 'LIB$EDFTYPE',
0093 'LIB$EDFSTRUCT',
0094
0095 )]
0096
0097 MODULE EDFVAR;
0098
0099 VAR
0100
0101 { +
0102 These are for the FDL file(s).
0103 - }
0104 FDL_DEST : [VOLATILE] TEXT;
0105 TT : TEXT;
0106 JOURNAL_FILE : [VOLATILE] TEXT;
0107 JOURNAL_ENABLED : [VOLATILE] BOOLEAN := FALSE;
0108 JOURNAL_FILENAME : STRING255;
0109 INPUT_FILENAME_DESC : DESCRIPTOR;
0110 OUTPUT_FILENAME_DESC : DESCRIPTOR;
0111 ANALYSIS_FILENAME_DESC : DESCRIPTOR;
0112 RES_OUTPUT_FILENAME_DESC : [VOLATILE] DESCRIPTOR;
0113 DEFAULT_FILENAME_DESC : DESCRIPTOR;
0114 NL_DEV_DESC : DESCRIPTOR;
0115 ANALYSIS_SPECIFIED : BOOLEAN := FALSE;
0116 ANALYSIS_ONLY : [VOLATILE] BOOLEAN := FALSE;
0117
0118 { +
0119 These are the main editing control variables.
0120 While editing is true, we stay in the editor.
0121 Take defaults is true if we're answering questions
0122 automatically.
0123 If no input is true, we skip the input fdl file operation.
0124 If auto tune is true, we don't do any output to the terminal,
0125 or expect any from it. It's all an automated Optimize script.
0126 - }
0127 EDITING : [VOLATILE] BOOLEAN := FALSE;
0128 TAKE_DEFAULTS : BOOLEAN := FALSE;
0129 NO_INPUT : BOOLEAN := FALSE;
0130 AUTO_TUNE : [VOLATILE] BOOLEAN := FALSE;
0131
0132 { +
0133 This is for outputting FDL to the terminal.
0134 - }
0135 SYSS$OUTPUT_NAME : [VOLATILE] STRING11 := 'SYSS$OUTPUT: ';
0136
0137 { +
0138 These indicate whether classes of errors were detected.
0139 - }
0140 SYSS$INPUT_ERROR : [VOLATILE] BOOLEAN := FALSE;
0141 RMS_INPUT_ERROR : [VOLATILE] BOOLEAN := FALSE;
```



```
0142 RMS_OUTPUT_ERROR : [VOLATILE] BOOLEAN := FALSE;
0143 CONTROL_ZEE_TYPED : [VOLATILE] BOOLEAN := FALSE;
0144 MAIN_CTRLZ : [VOLATILE] BOOLEAN := FALSE;
0145 MAIN_LEVEL : [VOLATILE] BOOLEAN := TRUE;
0146 QUESTION_TYPED : [VOLATILE] BOOLEAN := FALSE;
0147
0148 { +
0149 This is for graphing.
0150 - }
0151 XY_PLOT : XY_PLOT_TYPE;
0152 COLOR_PLOT : XY_PLOT_TYPE;
0153 XY_ARRAY : XY_ARRAY_TYPE;
0154 COLOR_ARRAY : XY_ARRAY_TYPE;
0155 COLOR_ROW : PACKED ARRAY [0..(BKT$C_MAXBKTSIZ-1)] OF BYTE;
0156 BREAKPOINT_LEFT : INTEGER;
0157 BREAKPOINT_MID : INTEGER;
0158 BREAKPOINT_RIGHT : INTEGER;
0159 DEPTHPOINT_LEFT : INTEGER;
0160 DEPTHPOINT_MID : INTEGER;
0161 DEPTHPOINT_RIGHT : INTEGER;
0162 EXAMPOINT_LEFT : INTEGER;
0163 EXAMPOINT_MID : INTEGER;
0164 EXAMPOINT_RIGHT : INTEGER;
0165 NUMPOINT_LEFT : INTEGER;
0166 NUMPOINT_MID : INTEGER;
0167 NUMPOINT_RIGHT : INTEGER;
0168 PAGEPOINT_LEFT : INTEGER;
0169 PAGEPOINT_MID : INTEGER;
0170 PAGEPOINT_RIGHT : INTEGER;
0171 GRAPH_TYPE : INTEGER;
0172 CURRENT_GRAPH_INDEX : INTEGER;
0173 LAST_GRAPH_INDEX : INTEGER;
0174 STEPS : INTEGER;
0175 Y_LABEL : STRING32;
0176
0177 { +
0178 These are the 'width' arrays that indicate how long a particular keyword
0179 should be typed.
0180 - }
0181 PRIMARY_WIDTH : PACKED ARRAY [PRIMARY_TYPE] OF BYTE;
0182 SECONDARY_WIDTH : PACKED ARRAY [SECONDARY_TYPE] OF BYTE;
0183
0184 { +
0185 This array stores the maximum value of the number-valued secondaries.
0186 - }
0187 SECONDARY_MAX : ARRAY [SECONDARY_TYPE] OF INTEGER;
0188
0189 { +
0190 This array stores the legal sequencing of Primaries as defined by the
0191 FDL Specification.
0192 - }
0193 PRI_SEQ : [VOLATILE] PACKED ARRAY [PRIMARY_TYPE] OF BYTE;
0194
0195 { +
0196 These store the character sequences to set the video attribute modes
0197 of the VT100 (and compatible) terminals.
0198 - }
```



```
0199 ANSI_RESET      : [VOLATILE] STRING4 := ('(27)', ['0', 'm']);
0200 ANSI_BOLD        : STRING4 := ('(27)', ['1', 'm']);
0201 ANSI_UNDERSCORE  : STRING4 := ('(27)', ['4', 'm']);
0202 ANSI_BLINK       : STRING4 := ('(27)', ['5', 'm']);
0203 ANSI_REVERSE     : [VOLATILE] STRING4 := ('(27)', ['7', 'm']);
0204
0205 VID_STRING4       : STRING4;
0206 NULL_STRING4      : STRING4 := ('(0)', '(0)', '(0)', '(0)');
0207 EMPTY_STRING     : STRING2 := ('', '');
0208 SHIFT             : [VOLATILE] STRING4 :=
0209 ('(9)', '(0)', '(0)', '(0)');
0210 CRLF_SHIFT        : [VOLATILE] STRING6 :=
0211 ('(13)', '(10)', '(9)', '(0)', '(0)', '(0)');
0212 LOW_SHIFT        : STRING3;
0213
0214 NULL_CHAR         : CHAR := ('(0)');
0215
0216 { +
0217 This is the terminal and screen database.
0218 - }
0219 TAB               : CHAR := ('(9)');
0220 ESCAPE            : [VOLATILE] CHAR := ('(27)');
0221 APOSTROPHE        : CHAR := ('(39)');
0222 CONTROL_W         : CHAR := ('(23)');
0223 CONTROL_Z         : [VOLATILE] CHAR := ('(26)');
0224 QUESTION_MARK    : [VOLATILE] CHAR := ('(63)');
0225 ERR_CHAR         : [VOLATILE] CHAR;
0226 CONTROL_G        : CHAR := ('(7)');
0227 CRLF             : [VOLATILE] STRING2 := ('(13)', '(10)');
0228 TERMINAL_TYPE    : BYTE;
0229 TERMINAL_SPEED    : INTEGER;
0230 ANSI_CRT         : BOOLEAN := FALSE;
0231 REGIS            : [VOLATILE] BOOLEAN := FALSE;
0232 DEC_CRT          : BOOLEAN := FALSE;
0233 DEV_TYPE         : INTEGER;
0234 VIDEO_TERMINAL   : BOOLEAN := FALSE;
0235 VID_TERM         : INTEGER;
0236 SCREEN_FLAGS     : [VOLATILE] SCR1$TYPE;
0237 OUT_LINE        : [VOLATILE] TEMP_VARYING;
0238 ONE             : [VOLATILE] INTEGER := 1;
0239 CHFFLAGS        : [VOLATILE] INTEGER := 0;
0240 FLAGS           : [VOLATILE] FDL2$TYPE;
0241 TEMP_FDL3$TYPE  : [VOLATILE] FDL3$TYPE;
0242 LINE_WIDTH      : INTEGER;
0243 LINES_PER_PAGE  : [VOLATILE] INTEGER;
0244 DEST_IS_TERMINAL : [VOLATILE] BOOLEAN;
0245 LINES_SHOWN     : [VOLATILE] INTEGER;
0246 MINIMUM_TERM_WIDTH : INTEGER := 80;
0247 MINIMUM_VIDEO_PAGE : INTEGER := 24;
0248 SCROLLING_SET   : [VOLATILE] BOOLEAN := FALSE;
0249 FULL_PROMPT     : BOOLEAN := TRUE;
0250 TEMP_FULL_PROMPT : [VOLATILE] BOOLEAN := FALSE;
0251 ORIG_TIME       : REAL;
0252 QUAD_TIME       : QUADWORD;
0253
0254 DEFAULT_PRIMARY  : PRIMARY_TYPE := FILES;
0255 DEFAULT_PRINUM   : INTEGER := 0;
```



```
0256 COL ONE           : INTEGER := 1;
0257 LINE ONE         : [VOLATILE] INTEGER := 1;
0258 LOWER_LINE       : INTEGER := 17;
0259 PROMPT_LINE       : INTEGER := 23;
0260
0261 PARAM_BLOCK        : [VOLATILE]TPASTYPE;
0262
0263 SEC_ATTR           : STRING22 := ' Secondary Attributes ';
0264 EDFHLP_STRING       : STRING6 := 'EDFHLP';
0265 IDENT_STRING        : STRING40 :=
0266                                     VAX-11 FDL Editor';
0267 IDENT_STRING_LENGTH : INTEGER := 40;
0268 QUES_HINT           : STRING31 := '(Type "?" for list of Keywords)';
0269 EDF_HEADER          : STRING19 := ' VAX-11 FDL Editor ';
0270 CONTINUE_TEXT       : STRING45 :=
0271                                     ' Press RETURN to continue (^Z for Main Menu) ';
0272 ISTATUS             : [VOLATILE] INTEGER;
0273 FAB_DUMMY           : FAB_PTR;
0274 RAB_DUMMY           : FAB_PTR;
0275
0276 FDL_BLOCK           : [VOLATILE] ^FDL_TYPE;
0277 FDL$AL_BLOCK        : [EXTERNAL,VOLATILE] INTEGER;
0278
0279 EDF$GL_SECNUM        : [EXTERNAL,VOLATILE] LONG;
0280 EDF$GL_PROT_MASK     : [EXTERNAL] CTRL_ARRAY;
0281 EDF$GL_FID1          : [EXTERNAL,VOLATILE] LONG;
0282 EDF$GL_FID2          : [EXTERNAL,VOLATILE] LONG;
0283 EDF$GL_FID3          : [EXTERNAL,VOLATILE] LONG;
0284 EDF$GL_OWNER_UIC     : [EXTERNAL,VOLATILE] LONG;
0285 EDF$GL_SPARET        : [EXTERNAL,VOLATILE] LONG;
0286 EDF$AB_STRING        : [EXTERNAL,VOLATILE] DESCRIPTOR;
0287 EDF$AB_COMMENT       : [EXTERNAL,VOLATILE] DESCRIPTOR;
0288 EDF$AB_UIC_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0289 EDF$AB_UIC_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0290 EDF$AB_FID_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0291 EDF$AB_FID_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0292 EDF$AB_PRIMARY_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0293 EDF$AB_PRIMARY_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0294 EDF$AB_ACCESS_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0295 EDF$AB_ACCESS_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0296 EDF$AB_ACL_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0297 EDF$AB_ACL_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0298 EDF$AB_AREA_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0299 EDF$AB_AREA_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0300 EDF$AB_CONNECT_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0301 EDF$AB_CONNECT_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0302 EDF$AB_DATE_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0303 EDF$AB_DATE_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0304 EDF$AB_FILE_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0305 EDF$AB_FILE_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0306 EDF$AB_JOURNAL_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0307 EDF$AB_JOURNAL_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0308 EDF$AB_KEY_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0309 EDF$AB_KEY_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0310 EDF$AB_RECORD_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;
0311 EDF$AB_RECORD_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
0312
```



```
0313 EDF$AB_SHARING_TABLE_KEY      : [EXTERNAL,VOLATILE] LONG;  
0314 EDF$AB_SHARING_TABLE_STA    : [EXTERNAL,VOLATILE] LONG;  
0315 EDF$AB_SYSTEM_TABLE_KEY     : [EXTERNAL,VOLATILE] LONG;  
0316 EDF$AB_SYSTEM_TABLE_STA    : [EXTERNAL,VOLATILE] LONG;  
0317 EDF$AB_POSIT_TABLE_KEY      : [EXTERNAL,VOLATILE] LONG;  
0318 EDF$AB_POSIT_TABLE_STA     : [EXTERNAL,VOLATILE] LONG;  
0319 EDF$AB_PROT_TABLE_KEY       : [EXTERNAL,VOLATILE] LONG;  
0320 EDF$AB_PROT_TABLE_STA      : [EXTERNAL,VOLATILE] LONG;  
0321 EDF$AB_ORG_TABLE_KEY        : [EXTERNAL,VOLATILE] LONG;  
0322 EDF$AB_ORG_TABLE_STA       : [EXTERNAL,VOLATILE] LONG;  
0323 EDF$AB_SOURCE_TABLE_KEY     : [EXTERNAL,VOLATILE] LONG;  
0324 EDF$AB_SOURCE_TABLE_STA    : [EXTERNAL,VOLATILE] LONG;  
0325 EDF$AB_RU_TABLE_KEY        : [EXTERNAL,VOLATILE] LONG;  
0326 EDF$AB_RU_TABLE_STA       : [EXTERNAL,VOLATILE] LONG;  
0327 EDF$AB_CARR_TABLE_KEY      : [EXTERNAL,VOLATILE] LONG;  
0328 EDF$AB_CARR_TABLE_STA     : [EXTERNAL,VOLATILE] LONG;  
0329 EDF$AB_FORMAT_TABLE_KEY    : [EXTERNAL,VOLATILE] LONG;  
0330 EDF$AB_FORMAT_TABLE_STA    : [EXTERNAL,VOLATILE] LONG;  
0331 EDF$AB_TYPE_TABLE_KEY       : [EXTERNAL,VOLATILE] LONG;  
0332 EDF$AB_TYPE_TABLE_STA      : [EXTERNAL,VOLATILE] LONG;  
0333 EDF$AB_LOAD_METHOD_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0334 EDF$AB_LOAD_METHOD_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0335 EDF$AB_WEIGHT_TABLE_KEY     : [EXTERNAL,VOLATILE] LONG;  
0336 EDF$AB_WEIGHT_TABLE_STA    : [EXTERNAL,VOLATILE] LONG;  
0337 EDF$AB_SURFACE_OPTION_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0338 EDF$AB_SURFACE_OPTION_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0339 EDF$AB_CURRENT_FUNC_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0340 EDF$AB_CURRENT_FUNC_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0341 EDF$AB_DESIGN_CYCLE_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0342 EDF$AB_DESIGN_CYCLE_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0343 EDF$AB_SCRIPT_OPTION_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0344 EDF$AB_SCRIPT_OPTION_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0345 EDF$AB_YES_NO_TABLE_KEY     : [EXTERNAL,VOLATILE] LONG;  
0346 EDF$AB_YES_NO_TABLE_STA    : [EXTERNAL,VOLATILE] LONG;  
0347 EDF$AB_SET_FUNCTION_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0348 EDF$AB_SET_FUNCTION_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0349 EDF$AB_GRANULARITY_TABLE_KEY : [EXTERNAL,VOLATILE] LONG;  
0350 EDF$AB_GRANULARITY_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0351 EDF$AB_PROMPTING_TABLE_KEY  : [EXTERNAL,VOLATILE] LONG;  
0352 EDF$AB_PROMPTING_TABLE_STA : [EXTERNAL,VOLATILE] LONG;  
0353 EDF$AB_RESPONSES_TABLE_KEY  : [EXTERNAL,VOLATILE] LONG;  
0354 EDF$AB_RESPONSES_TABLE_STA : [EXTERNAL,VOLATILE] LONG;
```

```
0355 ( +  
0356 The following are the pointers to the Definition Linked List.  
0357 - )
```

```
0359 DEF_CURRENT      : [VOLATILE] ^LINE_OBJECT := NIL;  
0360 DEF_SCRATCH      : [VOLATILE] ^LINE_OBJECT := NIL;  
0361 DEF_HEAD         : [VOLATILE] ^LINE_OBJECT := NIL;  
0362 DEF_TAIL         : [VOLATILE] ^LINE_OBJECT := NIL;  
0363 DEF_SUCC         : [VOLATILE] ^LINE_OBJECT := NIL;  
0364 DEF_PRED         : [VOLATILE] ^LINE_OBJECT := NIL;
```

```
0365 ( +  
0366 These point to the analysis linked list.  
0367 - )  
0368 DEF_ANL_HEAD     : ^LINE_OBJECT := NIL;  
0369
```



```
0370 DEF_ANL_TAIL      : ^LINE_OBJECT := NIL;
0371 DEF_SAVE_HEAD    : ^LINE_OBJECT := NIL;
0372 DEF_SAVE_TAIL     : ^LINE_OBJECT := NIL;
0373
0374 POINTING_AT_DEFINITION : BOOLEAN := TRUE;
0375
0376 FILE_CREATED       : [VOLATILE] BOOLEAN := FALSE;
0377
0378 { +
0379 These are used for input string processing.
0380 - }
0381 INPUT_DESC          : [VOLATILE] DESCRIPTOR;
0382 INPUT_STRING        : [VOLATILE] STRING255;
0383 INPUT_VALUE         : INTEGER;
0384 INPUT_NUMBER        : INTEGER;
0385
0386 QUAD_DESC           : [VOLATILE] RECORD
0387     CASE QWHICH : QD SWITCH OF
0388         QWORD    : (TWO LONG : PACKED RECORD
0389                     L1,L2 : LONG
0390                     END);
0391         DWORD    : (DSC : DESCRIPTOR)
0392     END;
0393
0394 { +
0395 ACTIVE CALCULATION DATABASE.
0396 - }
0397 LINKED               : BOOLEAN;
0398 ACTIVE_AREA          : INTEGER;
0399 ACTIVE_PRIMARY       : PRIMARY_TYPE;
0400 VARIABLE_RECORDS     : BOOLEAN;
0401 CUR_MAX_REC          : INTEGER;
0402 BYTES_PER_BUCKET     : INTEGER;
0403 BUCKET_DEFAULT       : INTEGER;
0404 PRIMARY_INDEX_BUCKETS : INTEGER;
0405 INIT_PRIMARY_BUCKETS : ARRAY [0..31] OF INTEGER;
0406 ADDED_PRIMARY_BUCKETS : ARRAY [0..31] OF INTEGER;
0407 INIT_NUMBER_BUCKETS  : ARRAY [0..31] OF INTEGER;
0408 ADDED_NUMBER_BUCKETS : ARRAY [0..31] OF INTEGER;
0409 RECS_PER_BUCKET      : ARRAY [0..31] OF INTEGER;
0410 DEEPEST              : INTEGER;
0411 FIRST_PLOT           : BOOLEAN;
0412 OPTIMIZING           : BOOLEAN;
0413 VISIBLE_QUESTION     : BOOLEAN;
0414 WAIT_HELP            : BOOLEAN;
```



```
0416 { +
0417 QTAB is the table that drives the Q+A routine - QUERY
0418 The xDATA arrays hold the main EDF database.
0419 - }
0420 QTAB      : ARRAY [EDFSK_QTABSTART..EDFSK_QTABEND] OF QTAB_ENTRY;
0421 QTAB_OFFSET : INTEGER;
0422
0423 { +
0424 String descriptor elements
0425 - }
0426 SDATA      : ARRAY [EDFSK_SDATASTART..EDFSK_SDATAEND] OF DESCRIPTOR;
0427
0428 { +
0429 Real elements
0430 - }
0431 RDATA      : ARRAY [EDFSK_RDATASTART..EDFSK_RDATAEND] OF REAL;
0432
0433 { +
0434 Boolean elements
0435 - }
0436 BDATA      : ARRAY [EDFSK_BDATASTART..EDFSK_BDATAEND] OF BOOLEAN;
0437
0438 { +
0439 Integer elements
0440 - }
0441 IDATA      : ARRAY [EDFSK_IDATASTART..EDFSK_IDATAEND] OF INTEGER;
0442
0443 { +
0444 Valid (boolean) elements
0445 - }
0446 VDATA      : ARRAY [EDFSK_VDATASTART..EDFSK_VDATAEND] OF BOOLEAN;
0447
0448 { +
0449 Misc. scratch variables used during the Q+A.
0450 - }
0451 TEMP_STRING255      : STRING255;
0452 TEMP_DESCRIPTOR     : [VOLATILE] DESCRIPTOR;
0453 QUERY_FLAG          : BOOLEAN;
0454 LOW_KEY              : INTEGER := 0;
0455 HIGH_KEY             : INTEGER := 0;
0456 LOW_AREA            : INTEGER := 0;
0457 HIGH_AREA           : INTEGER := 0;
0458 FOUND_AREA          : BOOLEAN := FALSE;
0459 FOUND_KEY            : BOOLEAN := FALSE;
0460 FOUND_O              : BOOLEAN := FALSE;
0461 MAX_KEY_SIZE        : INTEGER;
0462 MIN_KEY_SIZE        : INTEGER;
0463 SEGMENT_WANTED       : ARRAY [0..7] OF BOOLEAN :=
0464 (FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE);
0465 SEGMENT_POSITION     : ARRAY [0..7] OF INTEGER;
0466 SEGMENT_LENGTH       : ARRAY [0..7] OF INTEGER;
0467 SEGMENT_NUMBER       : INTEGER;
0468 BUCKET_OVERHEAD      : INTEGER;
0469 MIN_BUCKET           : INTEGER;
0470 ENTRY_SIZE           : INTEGER;
0471 LOWMAX               : INTEGER;
0472 EXTRA                : INTEGER;
```

```
0473 CUR_MAX_FIXED          : INTEGER;
0474 MAX_STRING_ANSWER_LENGTH : INTEGER;
0475 OLD_COUNT               : INTEGER;
0476 GLOBAL_SET              : BOOLEAN;
0477 NUMBER_KEYS_SET         : BOOLEAN := FALSE;
0478 ISAM_ORG                 : BOOLEAN;
0479 MAX_KEY_POSITION         : INTEGER;
0480 TEMP_REAL                : REAL;
0481 TEMP_STATUS              : INTEGER;
0482 TEMP_INT2                : INTEGER;
0483 DEF                      : INTEGER;
0484 NULL_STRING              : [VOLATILE]DESCRIPTOR;
0485 TEST                     : LINE_OBJECT;
0486 FULL_CHOICE              : BOOLEAN;
0487
0488 { +
0489 This is the array of secondary value types.
0490 - }
0491 SEC_TYPE                  : [VOLATILE] PACKED ARRAY [SECONDARY_TYPE] OF
0492                             PACKED RECORD
0493                             STR      : BOOLEAN;
0494                             NUM      : BOOLEAN;
0495                             QUAL     : BOOLEAN;
0496                             SW       : BOOLEAN;
0497                             END;
0498
0499 { +
0500 This is the template for line_objects generated by MAKE_SCRATCH.
0501 - }
0502 LINE_OBJECT_TEMPLATE      : [VOLATILE] LINE_OBJECT;
```



VALUE  
%INCLUDE 'SRCS:EDFVALUE.PAS'  
{ \*\*

FILE: SRCS:EDFVALUE.PAS - Pascal include file to define  
initial values of selected top-level variables.

```
*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

FACILITY: VAX/VMS EDF (EDIT/FDL) UTILITY

ABSTRACT: This facility is used to create, modify, and optimize  
FDL specification files.

ENVIRONMENT: NATIVE/USER MODE

AUTHOR: Ken F. Henderson Jr.

CREATION DATE: 27-Mar-1981

MODIFIED BY:

V03-009	KFH0009	Ken Henderson	10 Sep 1983
		Support named UICs.	
V03-008	KFH0008	Ken Henderson	9 Aug 1983
		Fix max value of CLUSTER_SIZE.	
		Fix default of QTAB[TEST_PRIMARY].	
V03-007	KFH0007	Ken Henderson	30 Jul 1983
		Fix SEC TYPE table for audit_trail.	
		Add DEFERRED_WRITE.	



EDFVAR  
V04-000

Source Listing

6 7  
16-Sep-1984 00:42:36  
15-Sep-1984 22:43:40

VAX-11 Pascal V2.4-277  
\_S255SDUA28:[EDF.SRC]EDFVALUE.PAS;1 (1) Page 12

0561  
0562  
0563  
0564  
0565  
0566  
0567  
0568  
0569  
0570  
0571  
0572  
0573  
0574  
0575  
0576  
0577  
0578  
0579  
0580  
0581  
0582  
0583  
0584  
0585

-- )

V03-006	KFH0006	Ken Henderson	26 Apr 1983
	Fix various defaults in QTAB. Transferred some initializations to the EDFVAR declarations.		
V03-005	KFH0005	Ken Henderson	14 Apr 1983
	Changed max bucket size to 63 from 65. Added ANALYSIS, OUTPUT, RESPONSES, PROMPTING, SET FUNCTION, GRANULARITY. Added support for SEGMENTED keys.		
V03-004	KFH0004	Ken Henderson	7 Mar 1983
	Changed max bucket size to 65 from 127.		
V03-003	KFH0003	Ken Henderson	11 Sept 1982
	Added initialization of VDATA and BDATA.		
V03-002	KFH0002	Ken Henderson	9 Sept 1982
	Added initialization of QTAB.		
V03-001	KFH0001	Ken Henderson	23-Mar-1982
	Took out reference to EDITFDL_STRING		



```
0587 { +
0588 Initialize the Boolean-array to all false.
0589 - }
0590 BDATA      := (
0591     FALSE,
0592     FALSE,
0593     FALSE,
0594     FALSE,
0595     FALSE,
0596     FALSE,
0597     FALSE,
0598     FALSE,
0599     FALSE,
0600     FALSE,
0601     FALSE,
0602     FALSE,
0603     FALSE,
0604     FALSE,
0605     FALSE,
0606     FALSE,
0607     FALSE,
0608     FALSE,
0609     FALSE,
0610     FALSE,
0611     FALSE,
0612     FALSE,
0613     FALSE,
0614     FALSE,
0615     FALSE,
0616 );
0617
0618 { +
0619 Initialize the String-array to all null-string.
0620 - }
0621 SDATA      := (
0622     (0,DSC$K_DTYPE-T,DSC$K_CLASS-D,NIL),
0623     (0,DSC$K_DTYPE-T,DSC$K_CLASS-D,NIL),
0624     (0,DSC$K_DTYPE-T,DSC$K_CLASS-D,NIL),
0625     (0,DSC$K_DTYPE-T,DSC$K_CLASS-D,NIL),
0626     (0,DSC$K_DTYPE-T,DSC$K_CLASS-D,NIL),
0627     (0,DSC$K_DTYPE-T,DSC$K_CLASS-D,NIL),
0628     (0,DSC$K_DTYPE-T,DSC$K_CLASS-D,NIL),
0629 );
0630
0631 { +
0632 Initialize the Valid-array to all false.
0633 - }
0634 VDATA      := (
0635     FALSE,
0636     FALSE,
0637     FALSE,
0638     FALSE,
0639     FALSE,
0640     FALSE,
0641     FALSE,
0642     FALSE,
0643     FALSE,
```

EDFVAR  
V04-000

Source Listing

1 7  
16-Sep-1984 00:42:36  
15-Sep-1984 22:43:40

VAX-11 Pascal V2.4-277  
\_S255SDUA28:[EDF.SRC]EDFVALUE.PAS;1 (2) Page 14

```
0644 1 FALSE,  
0645 1 FALSE,  
0646 1 FALSE,  
0647 1 FALSE,  
0648 1 FALSE,  
0649 1 FALSE,  
0650 1 FALSE,  
0651 1 FALSE,  
0652 1 FALSE,  
0653 1 FALSE,  
0654 1 FALSE,  
0655 1 FALSE,  
0656 1 FALSE,  
0657 1 FALSE,  
0658 1 FALSE,  
0659 1 FALSE,  
0660 1 FALSE,  
0661 1 FALSE,  
0662 1 FALSE,  
0663 1 FALSE,  
0664 1 FALSE,  
0665 1 FALSE,  
0666 1 FALSE,  
0667 1 FALSE,  
0668 1 FALSE,  
0669 1 FALSE,  
0670 1 FALSE,  
0671 1 FALSE,  
0672 1 FALSE,  
0673 1 FALSE,  
0674 1 FALSE,  
0675 1 FALSE,  
0676 1 FALSE,  
0677 1 FALSE,  
0678 1 FALSE,  
0679 1 FALSE,  
0680 1 FALSE,  
0681 1 FALSE,  
0682 1 FALSE,  
0683 1 FALSE,  
0684 1 FALSE,  
0685 1 FALSE,  
0686 1 FALSE,  
0687 1 FALSE,  
0688 1 FALSE,  
0689 1 FALSE,  
0690 1 FALSE,  
0691 1 FALSE,  
0692 1 FALSE,  
0693 1 FALSE,  
0694 1 FALSE,  
0695 1 FALSE,  
0696 1 FALSE,  
0697 1 FALSE,  
0698 1 FALSE,  
0699 1  
0700 1
```



```
0701 { +
0702 Initialize the sequencing array.
0703 - }
0704 PRI_SEQ := (
0705     15,      { DUMMY_PRIMARY$ }
0706     8,        { ACCESS, }
0707     4,        { ACL, }
0708     13,      { ANALYSIS_OF_AREA, }
0709     14,      { ANALYSIS_OF_KEY, }
0710     11,      { AREA, }
0711     10,      { CONNECT, }
0712     4,        { DATE, }
0713     3,        { FILE$, }
0714     1,        { IDENT, }
0715     6,        { JOURNAL, }
0716     12,      { KEY, }
0717     7,        { RECORD$, }
0718     9,        { SHARING, }
0719     2,        { SYSTEM, }
0720     0,        { TITLE }
0721 );
0722
0723 { +
0724 Initialize the 'width' arrays - that indicate how long a particular
0725 keyword should be printed.
0726 - }
0727 PRIMARY_WIDTH := (
0728     0,      { DUMMY_PRIMARY$ }
0729     6,      { ACCESS, }
0730     3,      { ACL, }
0731     16,     { ANALYSIS_OF_AREA, }
0732     15,     { ANALYSIS_OF_KEY, }
0733     4,      { AREA, }
0734     7,      { CONNECT, }
0735     4,      { DATE, }
0736     4,      { FILE$, }
0737     5,      { IDENT, }
0738     7,      { JOURNAL, }
0739     3,      { KEY, }
0740     6,      { RECORD$, }
0741     7,      { SHARING, }
0742     6,      { SYSTEM, }
0743     5,      { TITLE }
0744 );
0745
0746 SECONDARY_WIDTH := (
0747
0748 { RESERVE 0 } 0,      { DUMMY_SECONDARY$, }
0749
0750 { ACCESS PRIMARY }
0751
0752     8,      { BLOCK_IO$ }
0753     6,      { DELETES }
0754     3,      { GET$ }
0755     3,      { PUT$ }
```

```
0758 I      9,      { RECORD_IOS }
0759 I      8,      { TRUNCATES }
0760 I      6,      { UPDATES }
0761 I
0762 I { ACL PRIMARY }
0763 I
0764 I      5,      { ENTRY }
0765 I
0766 I { ANALYSIS_OF_AREA PRIMARY }
0767 I      15,     { RECLAIMED_SPACE }
0768 I
0769 I { ANALYSIS_OF_KEY PRIMARY }
0770 I
0771 I      9,      { DATA_FILLS, }
0772 I      20,     { DATA_KEY_COMPRESSION, }
0773 I      23,     { DATA_RECORD_COMPRESSION, }
0774 I      17,     { DATA_RECORD_COUNT, }
0775 I      19,     { DATA_SPACE_OCCUPIED, }
0776 I      9,      { DELETIONS, }
0777 I      5,      { DEPTH, }
0778 I      19,     { DUPLICATES_PER_SIDR, }
0779 I      17,     { INDEX_COMPRESSION, }
0780 I      10,     { INDEX_FILLS, }
0781 I      20,     { INDEX_SPACE_OCCUPIED, }
0782 I      19,     { LEVELT_RECORD_COUNT }
0783 I      16,     { MEAN_DATA_LENGTH, }
0784 I      17,     { MEAN_INDEX_LENGTH, }
0785 I      15,     { RANDOM_ACCESSES, }
0786 I      14,     { RANDOM_INSERTS, }
0787 I      19,     { SEQUENTIAL_ACCESSES, }
0788 I
0789 I { AREA PRIMARY }
0790 I
0791 I      10,     { ALLOCATIONS, }
0792 I      19,     { BEST_TRY_CONTIGUOUS$, }
0793 I      11,     { BUCKET_SIZES, }
0794 I      10,     { CONTIGUOUS$, }
0795 I      17,     { EXACT_POSITIONINGS$, }
0796 I      9,      { EXTENSIONS, }
0797 I      8,      { POSITIONS, }
0798 I      6,      { VOLUMES, }
0799 I
0800 I { CONNECT PRIMARY }
0801 I
0802 I      12,     { ASYNCHRONOUS }
0803 I      8,      { BLOCK_IO }
0804 I      11,     { BUCKET_CODE }
0805 I      7,      { CONTEXT }
0806 I      11,     { END_OF_FILE }
0807 I      12,     { FILE_BUCKETS }
0808 I      11,     { FAST_DELETE }
0809 I      16,     { KEY_OF_REFERENCE }
0810 I      17,     { KEY_GREATER_EQUAL }
0811 I      16,     { KEY_GREATER_THAN }
0812 I      9,      { KEY_LIMIT }
0813 I      11,     { LOCATE_MODE }
0814 I      12,     { LOCK_ON_READ }
```



0815	I	13.	{ LOCK ON WRITE }
0816	I	16.	{ MANUAL_UNLOCKING }
0817	I	16.	{ MULTIBLOCK_COUNT }
0818	I	17.	{ MULTIBUFFER_COUNT }
0819	I	6.	{ NOLOCK }
0820	I	18.	{ NONEXISTENT_RECORD }
0821	I	10.	{ READ_AHEAD }
0822	I	15.	{ READ-REGARDLESS }
0823	I	14.	{ TIMEOUT_ENABLE }
0824	I	14.	{ TIMEOUT_PERIOD }
0825	I	15.	{ TRUNCATE ON PUT }
0826	I	19.	{ TT_CANCEL_CONTROL_0 }
0827	I	15.	{ TT_UPCASE_INPUT }
0828	I	9.	{ TT_PROMPT }
0829	I	19.	{ TT_PURGE_TYPE_AHEAD }
0830	I	14.	{ TT_READ_NOECHO }
0831	I	16.	{ TT_READ_NOFILTER }
0832	I	9.	{ UPDATE_IF }
0833	I	15.	{ WAIT_FOR_RECORD }
0834	I	12.	{ WRITE_BEHIND }
0835	I		
0836	I		{ DATE PRIMARY }
0837	I		
0838	I	6.	{ BACKUPS, }
0839	I	8.	{ CREATIONS, }
0840	I	10.	{ EXPIRATIONS, }
0841	I	8.	{ REVISIONS, }
0842	I		
0843	I		{ FILE PRIMARY }
0844	I		
0845	I	10.	{ ALLOCATION, }
0846	I	19.	{ BEST_TRY_CONTIGUOUS, }
0847	I	11.	{ BUCKET_SIZE, }
0848	I	12.	{ CLUSTER_SIZE, }
0849	I	7.	{ CONTEXTS }
0850	I	10.	{ CONTIGUOUS, }
0851	I	9.	{ CREATE_IF }
0852	I	12.	{ DEFAULT_NAME, }
0853	I	14.	{ DEFERRED_WRITE, }
0854	I	15.	{ DELETE_ON_CLOSE, }
0855	I	15.	{ DIRECTORY_ENTRY, }
0856	I	15.	{ ERASE_ON_DELETE, }
0857	I	9.	{ EXTENSION, }
0858	I	19.	{ GLOBAL_BUFFER_COUNT, }
0859	I	13.	{ MT_BLOCK_SIZE, }
0860	I	19.	{ MT_CURRENT_POSITION, }
0861	I	10.	{ MT_NOT_EOF }
0862	I	13.	{ MT_PROTECTION, }
0863	I	14.	{ MT_OPEN_REWIND, }
0864	I	15.	{ MT_CLOSE_REWIND }
0865	I	17.	{ MAX_RECORD_NUMBER, }
0866	I	16.	{ MAXIMIZE_VERSION, }
0867	I	4.	{ NAME, }
0868	I	8.	{ NOBACKUP, }
0869	I	19.	{ NON_FILE_STRUCTURED }
0870	I	17.	{ OUTPUT_FILE_PARSE }
0871	I	12.	{ ORGANIZATION, }

```
0872 I      5,      { OWNER, }
0873 I      14,     { PRINT_ON_CLOSE, }
0874 I      10,     { PROTECTION, }
0875 I      10,     { READ_CHECK, }
0876 I      8,      { REVISION, }
0877 I      15,     { SEQUENTIAL_ONLY }
0878 I      15,     { SUBMIT_ON_CLOSE, }
0879 I      9,      { SUPERSEDE, }
0880 I      9,      { TEMPORARY }
0881 I      17,     { TRUNCATE_ON_CLOSE, }
0882 I      14,     { USER_FILE_OPEN }
0883 I      11,     { WINDOW_SIZE }
0884 I      11,     { WRITE_CHECK, }
0885 I
0886 I      { JOURNALING PRIMARY }
0887 I
0888 I      11,     { AFTER_IMAGE }
0889 I      10,     { AFTER_NAME }
0890 I      11,     { AUDIT-TRAIL }
0891 I      10,     { AUDIT-NAME }
0892 I      12,     { BEFORE_IMAGE }
0893 I      11,     { BEFORE-NAME }
0894 I      13,     { RECOVERY_UNIT, }
0895 I
0896 I      { KEY PRIMARY }
0897 I
0898 I      7,      { CHANGES, }
0899 I      9,      { DATA_AREA, }
0900 I      9,      { DATA_FILL, }
0901 I      20,     { DATA_KEY_COMPRESSION, }
0902 I      23,     { DATA_RECORD_COMPRESSION, }
0903 I      10,     { DUPLICATES, }
0904 I      10,     { INDEX_AREA, }
0905 I      17,     { INDEX_COMPRESSION, }
0906 I      10,     { INDEX_FILL, }
0907 I      17,     { LEVELT_INDEX_AREA, }
0908 I      4,      { NAMES, }
0909 I      8,      { NULL_KEY, }
0910 I      10,     { NULL_VALUE, }
0911 I      6,      { PROLOG(UE) - 1ST 6 CHARS ONLY }
0912 I      0,      { SEG_LENGTH, }
0913 I      0,      { SEG_POSITION, }
0914 I      0,      { SEG_TYPE, }
0915 I
0916 I      { RECORD PRIMARY }
0917 I
0918 I      10,     { BLOCK_SPAN, }
0919 I      16,     { CARRIAGE_CONTROL, }
0920 I      18,     { CONTROL_FIELD_SIZE, }
0921 I      6,      { FORMAT, }
0922 I      4,      { SIZE, }
0923 I
0924 I      { SHARING PRIMARY }
0925 I
0926 I      6,      { DELETE }
0927 I      3,      { GET }
0928 I      11,     { MULTISTREAM }
```



```
0929 I      8,      { PROHIBIT }
0930 I      3,      { PUT }
0931 I      6,      { UPDATE }
0932 I     14,      { USER_INTERLOCK }
0933 I
0934 I { SYSTEM PRIMARY }
0935 I
0936 I      6,      { DEVICE, }
0937 I      6,      { SOURCE, }
0938 I      6,      { TARGET, }
0939 I
0940 I      );
0941 I
0942 I { +
0943 I   These are the maximum values of number-valued secondaries.
0944 I - }
0945 I
0946 I SECONDARY_MAX := (
0947 I { RESERVE 0 }      0,      { DUMMY_SECONDARY$, }
0948 I { ACCESS PRIMARY }
0949 I
0950 I      0,      { BLOCK_IO$ }
0951 I      0,      { DELETES }
0952 I      0,      { GET$ }
0953 I      0,      { PUT$ }
0954 I      0,      { RECORD_IO$ }
0955 I      0,      { TRUNCATES }
0956 I      0,      { UPDATES }
0957 I
0958 I { ACL PRIMARY }
0959 I
0960 I      0,      { ENTRY }
0961 I
0962 I { ANALYSIS_OF_AREA PRIMARY }
0963 I      0,      { RECLAIMED_SPACE }
0964 I
0965 I { ANALYSIS_OF_KEY PRIMARY }
0966 I
0967 I      0,      { DATA_FILLS, }
0968 I      0,      { DATA_KEY_COMPRESSION, }
0969 I      0,      { DATA_RECORD_COMPRESSION, }
0970 I      0,      { DATA_RECORD_COUNT, }
0971 I      0,      { DATA_SPACE_OCCUPIED, }
0972 I      0,      { DELETIONS, }
0973 I      0,      { DEPTH, }
0974 I      0,      { DUPLICATES_PER_SIDR, }
0975 I      0,      { INDEX_COMPRESSION, }
0976 I      0,      { INDEX_FILLS, }
0977 I      0,      { INDEX_SPACE_OCCUPIED, }
0978 I      0,      { LEVELT_RECORD_COUNT }
0979 I      0,      { MEAN_DATA_LENGTH, }
0980 I      0,      { MEAN_INDEX_LENGTH, }
0981 I      0,      { RANDOM_ACCESSES, }
0982 I      0,      { RANDOM_INSERTS, }
0983 I      0,      { SEQUENTIAL_ACCESSES, }
0984 I
0985 I
```

```
0986 I
0987 I      ( AREA PRIMARY )
0988 I
0989 I      EDF$C_1GIGA, ( ALLOCATIONS, )
0990 I      0,           ( BEST_TRY_CONTIGUOUS, )
0991 I      BKT$C_MAXBKTSIZ, ( BUCKET_SIZES, )
0992 I      0,           ( CONTIGUOUS, )
0993 I      0,           ( EXACT_POSITIONINGS, )
0994 I      EDF$C_1GIGA, ( EXTENSIONS, )
0995 I      16777215,    ( POSITIONS, )
0996 I      65535,       ( VOLUMES, )
0997 I
0998 I      ( CONNECT PRIMARY )
0999 I
1000 I      0,           ( ASYNCHRONOUS )
1001 I      0,           ( BLOCK_IO )
1002 I      EDF$C_1GIGA, ( BUCKET_CODE )
1003 I      EDF$C_1GIGA, ( CONTEXT )
1004 I      0,           ( END_OF_FILE )
1005 I      0,           ( FILE_BUCKETS )
1006 I      0,           ( FAST_DELETE )
1007 I      255,         ( KEY_OF_REFERENCE )
1008 I      0,           ( KEY_GREATER_EQUAL )
1009 I      0,           ( KEY_GREATER_THAN )
1010 I      0,           ( KEY_LIMIT )
1011 I      0,           ( LOCATE_MODE )
1012 I      0,           ( LOCK_ON_READ )
1013 I      0,           ( LOCK_ON_WRITE )
1014 I      0,           ( MANUAL_UNLOCKING )
1015 I      255,         ( MULTIBLOCK_COUNT )
1016 I      255,         ( MULTIBUFFER_COUNT )
1017 I      0,           ( NOLOCK )
1018 I      0,           ( NONEXISTENT_RECORD )
1019 I      0,           ( READ_AHEAD )
1020 I      0,           ( READ_REGARDLESS )
1021 I      0,           ( TIMEOUT_ENABLE )
1022 I      255,         ( TIMEOUT_PERIOD )
1023 I      0,           ( TRUNCATE_ON_PUT )
1024 I      0,           ( TT_CANCEL_CONTROL_0 )
1025 I      0,           ( TT_UPCASE_INPUT )
1026 I      0,           ( TT_PROMPT )
1027 I      0,           ( TT_PURGE_TYPE_AHEAD )
1028 I      0,           ( TT_READ_NOECHO )
1029 I      0,           ( TT_READ_NOFILTER )
1030 I      0,           ( UPDATE_IF )
1031 I      0,           ( WAIT_FOR_RECORD )
1032 I      0,           ( WRITE_BEHIND )
1033 I
1034 I      ( DATE PRIMARY )
1035 I
1036 I      0,           ( BACKUPS, )
1037 I      0,           ( CREATIONS, )
1038 I      0,           ( EXPIRATIONS, )
1039 I      0,           ( REVISIONS, )
1040 I
1041 I      ( FILE PRIMARY )
1042 I
```



EDFVAR  
V04-000

Source Listing

C 8  
16-Sep-1984 00:42:36  
15-Sep-1984 22:43:40

VAX-11 Pascal V2.4-277  
\_S255SDUA28:[EDF.SRC]EDFVALUE.PAS;1 (2) Page 21

```
1043 I EDFSC_1GIGA,{ ALLOCATION, }
1044 I 0,{ BEST_TRY_CONTIGUOUS, }
1045 I BKTSC_MAXBKTSIZ,{ BUCKET_SIZE, }
1046 I EDFSC_1GIGA,{ CLUSTER_SIZE, }
1047 I EDFSC_1GIGA,{ CONTEXT }
1048 I 0,{ CONTIGUOUS, }
1049 I 0,{ CREATE_IF }
1050 I 0,{ DEFAULT_NAME, }
1051 I 0,{ DEFERRED_WRITE, }
1052 I 0,{ DELETE_ON_CLOSE, }
1053 I 0,{ DIRECTORY_ENTRY, }
1054 I 0,{ ERASE_ON_DELETE, }
1055 I EDFSC_1GIGA,{ EXTENSION, }
1056 I EDFSC_MAX_GBL_BUFS,{ GLOBAL_BUFFER_COUNT, }
1057 I 65532,{ MT_BLOCK_SIZE, }
1058 I 0,{ MT_CURRENT_POSITION, }
1059 I 0,{ MT_NOT_EOF }
1060 I 0,{ MT_PROTECTION, }
1061 I 0,{ MT_OPEN_REWIND, }
1062 I 0,{ MT_CLOSE_REWIND }
1063 I EDFSC_1GIGA,{ MAX_RECORD_NUMBER, }
1064 I 0,{ MAXIMIZE_VERSION, }
1065 I 0,{ NAME, }
1066 I 0,{ NOBACKUP, }
1067 I 0,{ NON_FILE_STRUCTURED }
1068 I 0,{ OUTPUT_FILE_PARSE }
1069 I 0,{ ORGANIZATION, }
1070 I 0,{ OWNER, }
1071 I 0,{ PRINT_ON_CLOSE, }
1072 I 0,{ PROTECTION, }
1073 I 0,{ READ_CHECK, }
1074 I 65535,{ REVISION, }
1075 I 0,{ SEQUENTIAL_ONLY }
1076 I 0,{ SUBMIT_ON_CLOSE, }
1077 I 0,{ SUPERSEDE, }
1078 I 0,{ TEMPORARY }
1079 I 0,{ TRUNCATE_ON_CLOSE, }
1080 I 0,{ USER_FILE_OPEN }
1081 I EDFSC_1GIGA,{ WINDOW_SIZE }
1082 I 0,{ WRITE_CHECK, }
1083 I
1084 I { JOURNALING PRIMARY }
1085 I
1086 I 0,{ AFTER_IMAGE, }
1087 I 0,{ AFTER_NAME }
1088 I 0,{ AUDIT-TRAIL, }
1089 I 0,{ AUDIT-NAME }
1090 I 0,{ BEFORE_IMAGE, }
1091 I 0,{ BEFORE-NAME }
1092 I 0,{ RECOVERY_UNIT, }
1093 I
1094 I { KEY PRIMARY }
1095 I
1096 I 0,{ CHANGES, }
1097 I 254,{ DATA-AREA, }
1098 I 100,{ DATA-FILL, }
1099 I 99,{ DATA-KEY-COMPRESSION, }
```

```
1100 I          99,      { DATA_RECORD_COMPRESSION, }
1101 I          0,      { DUPLICATES, }
1102 I          254,    { INDEX_AREA, }
1103 I          99,     { INDEX_COMPRESSION, }
1104 I          100,    { INDEX_FILL, }
1105 I          254,    { LEVEL_INDEX_AREA, }
1106 I          0,      { NAMES, }
1107 I          0,      { NULL_KEY, }
1108 I          255,     { NULL_VALUE, }
1109 I          3,       { PROLOGUE }
1110 I          255,     { SEG_LENGTH, }
1111 I          16299,   { SEG_POSITION, }
1112 I          0,      { SEG_TYPE, }
1113 I
1114 I      { RECORD PRIMARY }
1115 I
1116 I          0,      { BLOCK_SPAN, }
1117 I          0,      { CARRIAGE_CONTROL, }
1118 I          255,     { CONTROL_FIELD_SIZE, }
1119 I          0,      { FORMAT, }
1120 I      EDF$K_MAXRECSIZ, { SIZE, }
1121 I
1122 I      { SHARING PRIMARY }
1123 I
1124 I          0,      { DELETE }
1125 I          0,      { GET }
1126 I          0,      { MULTISTREAM }
1127 I          0,      { PROHIBIT }
1128 I          0,      { PUT }
1129 I          0,      { UPDATE }
1130 I          0,      { USER_INTERLOCK }
1131 I
1132 I      { SYSTEM PRIMARY }
1133 I
1134 I          0,      { DEVICE, }
1135 I          0,      { SOURCE, }
1136 I          0,      { TARGET, }
1137 I
1138 I      );
1139 I
1140 I      { +
1141 I      These are the secondary value types.
1142 I      - }
1143 I
1144 I      SEC_TYPE := (
1145 I
1146 I      { +
1147 I      KEY:          STR, NUM, QUAL, SW
1148 I      - }
1149 I
1150 I      { RESERVE 0 }      (FALSE,FALSE,FALSE,FALSE),      { DUMMY_SECONDARY$, }
1151 I
1152 I      { ACCESS PRIMARY }
1153 I
1154 I          (FALSE,FALSE,FALSE,TRUE),      { BLOCK_IOS }
1155 I          (FALSE,FALSE,FALSE,TRUE),      { DELETES }
1156 I          (FALSE,FALSE,FALSE,TRUE),      { GETS }
```



```
1157 I      (FALSE,FALSE,FALSE,TRUE),      { PUT$ }
1158 I      (FALSE,FALSE,FALSE,TRUE),      { RECORD_IOS }
1159 I      (FALSE,FALSE,FALSE,TRUE),      { TRUNCATES }
1160 I      (FALSE,FALSE,FALSE,TRUE),      { UPDATES }
1161 I
1162 I      { ACL PRIMARY }
1163 I
1164 I      (TRUE,FALSE,FALSE,FALSE),      { ENTRY }
1165 I
1166 I      { ANALYSIS_OF_AREA PRIMARY }
1167 I
1168 I      (FALSE,FALSE,FALSE,FALSE),      { RECLAIMED_SPACE }
1169 I
1170 I      { ANALYSIS_OF_KEY PRIMARY }
1171 I
1172 I      (FALSE,FALSE,FALSE,FALSE),      { DATA_FILLS, }
1173 I      (FALSE,FALSE,FALSE,FALSE),      { DATA_KEY_COMPRESSION, }
1174 I      (FALSE,FALSE,FALSE,FALSE),      { DATA_RECORD_COMPRESSION, }
1175 I      (FALSE,FALSE,FALSE,FALSE),      { DATA_RECORD_COUNT, }
1176 I      (FALSE,FALSE,FALSE,FALSE),      { DATA_SPACE_OCCUPIED, }
1177 I      (FALSE,FALSE,FALSE,FALSE),      { DELETIONS, }
1178 I      (FALSE,FALSE,FALSE,FALSE),      { DEPTH, }
1179 I      (FALSE,FALSE,FALSE,FALSE),      { DUPLICATES_PER_SIDR, }
1180 I      (FALSE,FALSE,FALSE,FALSE),      { INDEX_COMPRESSION, }
1181 I      (FALSE,FALSE,FALSE,FALSE),      { INDEX_FILLS, }
1182 I      (FALSE,FALSE,FALSE,FALSE),      { INDEX_SPACE_OCCUPIED, }
1183 I      (FALSE,FALSE,FALSE,FALSE),      { LEVELT_RECORD_COUNT }
1184 I      (FALSE,FALSE,FALSE,FALSE),      { MEAN_DATA_LENGTH, }
1185 I      (FALSE,FALSE,FALSE,FALSE),      { MEAN_INDEX_LENGTH, }
1186 I      (FALSE,FALSE,FALSE,FALSE),      { RANDOM_ACCESSES, }
1187 I      (FALSE,FALSE,FALSE,FALSE),      { RANDOM_INSERTS, }
1188 I      (FALSE,FALSE,FALSE,FALSE),      { SEQUENTIAL_ACCESSES, }
1189 I
1190 I      { AREA PRIMARY }
1191 I
1192 I      (FALSE,TRUE,FALSE,FALSE),      { ALLOCATIONS, }
1193 I      (FALSE,FALSE,FALSE,TRUE),      { BEST_TRY_CONTIGUOUS, }
1194 I      (FALSE,TRUE,FALSE,FALSE),      { BUCKET_SIZES, }
1195 I      (FALSE,FALSE,FALSE,TRUE),      { CONTIGUOUS, }
1196 I      (FALSE,FALSE,FALSE,TRUE),      { EXACT_POSITIONINGS, }
1197 I      (FALSE,TRUE,FALSE,FALSE),      { EXTENSIONS, }
1198 I      (FALSE,FALSE,FALSE,FALSE),      { POSITIONS, }
1199 I      (FALSE,TRUE,FALSE,FALSE),      { VOLUMES, }
1200 I
1201 I      { +
1202 I      KEY:          STR,  NUM,  QUAL,  SW
1203 I      - }
1204 I
1205 I      { CONNECT PRIMARY }
1206 I
1207 I      (FALSE,FALSE,FALSE,TRUE),      { ASYNCHRONOUS }
1208 I      (FALSE,FALSE,FALSE,TRUE),      { BLOCK_IO }
1209 I      (FALSE,TRUE,FALSE,FALSE),      { BUCKET_CODE }
1210 I      (FALSE,TRUE,FALSE,FALSE),      { CONTEXT }
1211 I      (FALSE,FALSE,FALSE,TRUE),      { END_OF_FILE }
1212 I      (FALSE,FALSE,FALSE,TRUE),      { FILE_BUCKETS }
1213 I      (FALSE,FALSE,FALSE,TRUE),      { FAST_DELETE }
```

1214	I	(FALSE,TRUE,FALSE,FALSE),	{ KEY_OF_REFERENCE }
1215	I	(FALSE,FALSE,FALSE,TRUE),	{ KEY_GREATER_EQUAL }
1216	I	(FALSE,FALSE,FALSE,TRUE),	{ KEY_GREATER_THAN }
1217	I	(FALSE,FALSE,FALSE,TRUE),	{ KEY_LIMIT }
1218	I	(FALSE,FALSE,FALSE,TRUE),	{ LOCATE_MODE }
1219	I	(FALSE,FALSE,FALSE,TRUE),	{ LOCK_ON_READ }
1220	I	(FALSE,FALSE,FALSE,TRUE),	{ LOCK_ON_WRITE }
1221	I	(FALSE,FALSE,FALSE,TRUE),	{ MANUAL_UNLOCKING }
1222	I	(FALSE,TRUE,FALSE,FALSE),	{ MULTIBLOCK_COUNT }
1223	I	(FALSE,TRUE,FALSE,FALSE),	{ MULTIBUFFER_COUNT }
1224	I	(FALSE,FALSE,FALSE,TRUE),	{ NOLOCK }
1225	I	(FALSE,FALSE,FALSE,TRUE),	{ NONEXISTENT_RECORD }
1226	I	(FALSE,FALSE,FALSE,TRUE),	{ READ_AHEAD }
1227	I	(FALSE,FALSE,FALSE,TRUE),	{ READ_REGARDLESS }
1228	I	(FALSE,FALSE,FALSE,TRUE),	{ TIMEOUT_ENABLE }
1229	I	(FALSE,TRUE,FALSE,FALSE),	{ TIMEOUT_PERIOD }
1230	I	(FALSE,FALSE,FALSE,TRUE),	{ TRUNCATE_ON_PUT }
1231	I	(FALSE,FALSE,FALSE,TRUE),	{ TT_CANCEL_CONTROL_0 }
1232	I	(FALSE,FALSE,FALSE,TRUE),	{ TT_UPCASE_INPUT }
1233	I	(FALSE,FALSE,FALSE,TRUE),	{ TT_PROMPT }
1234	I	(FALSE,FALSE,FALSE,TRUE),	{ TT_PURGE_TYPE_AHEAD }
1235	I	(FALSE,FALSE,FALSE,TRUE),	{ TT_READ_NOECHO }
1236	I	(FALSE,FALSE,FALSE,TRUE),	{ TT_READ_NOFILTER }
1237	I	(FALSE,FALSE,FALSE,TRUE),	{ UPDATE_IF }
1238	I	(FALSE,FALSE,FALSE,TRUE),	{ WAIT_FOR_RECORD }
1239	I	(FALSE,FALSE,FALSE,TRUE),	{ WRITE_BEHIND }
1240	I		
1241	I	{ DATE_PRIMARY }	
1242	I		
1243	I	(TRUE,FALSE,FALSE,FALSE),	{ BACKUPS, }
1244	I	(TRUE,FALSE,FALSE,FALSE),	{ CREATIONS, }
1245	I	(TRUE,FALSE,FALSE,FALSE),	{ EXPIRATIONS, }
1246	I	(TRUE,FALSE,FALSE,FALSE),	{ REVISIONS, }
1247	I		
1248	I	{ FILE_PRIMARY }	
1249	I		
1250	I	(FALSE,TRUE,FALSE,FALSE),	{ ALLOCATION, }
1251	I	(FALSE,FALSE,FALSE,TRUE),	{ BEST_TRY_CONTIGUOUS, }
1252	I	(FALSE,TRUE,FALSE,FALSE),	{ BUCKET_SIZE, }
1253	I	(FALSE,TRUE,FALSE,FALSE),	{ CLUSTER_SIZE, }
1254	I	(FALSE,TRUE,FALSE,FALSE),	{ CONTEXTS }
1255	I	(FALSE,FALSE,FALSE,TRUE),	{ CONTIGUOUS, }
1256	I	(FALSE,FALSE,FALSE,TRUE),	{ CREATE_IF }
1257	I	(TRUE,FALSE,FALSE,FALSE),	{ DEFAULT_NAME, }
1258	I	(FALSE,FALSE,FALSE,TRUE),	{ DEFERRED_WRITE, }
1259	I	(FALSE,FALSE,FALSE,TRUE),	{ DELETE_ON_CLOSE, }
1260	I	(FALSE,FALSE,FALSE,TRUE),	{ DIRECTORY_ENTRY, }
1261	I	(FALSE,FALSE,FALSE,TRUE),	{ ERASE_ON_DELETE, }
1262	I	(FALSE,TRUE,FALSE,FALSE),	{ EXTENSION, }
1263	I	(FALSE,TRUE,FALSE,FALSE),	{ GLOBAL_BUFFER_COUNT, }
1264	I	(FALSE,TRUE,FALSE,FALSE),	{ MT_BLOCK_SIZE, }
1265	I	(FALSE,FALSE,FALSE,TRUE),	{ MT_CURRENT_POSITION, }
1266	I	(FALSE,FALSE,FALSE,TRUE),	{ MT_NOT_EOF }
1267	I	(FALSE,FALSE,FALSE,FALSE),	{ MT_PROTECTION, }
1268	I	(FALSE,FALSE,FALSE,TRUE),	{ MT_OPEN_REWIND, }
1269	I	(FALSE,FALSE,FALSE,TRUE),	{ MT_CLOSE_REWIND }
1270	I	(FALSE,TRUE,FALSE,FALSE),	{ MAX_RECORD_NUMBER, }



```
1271 I      (FALSE,FALSE,FALSE,TRUE),
1272 I      (TRUE,FALSE,FALSE,FALSE),
1273 I      (FALSE,FALSE,FALSE,TRUE),
1274 I      (FALSE,FALSE,FALSE,TRUE),
1275 I      (FALSE,FALSE,FALSE,TRUE),
1276 I      (FALSE,FALSE,TRUE,FALSE),
1277 I      (FALSE,FALSE,FALSE,FALSE),
1278 I      (FALSE,FALSE,FALSE,TRUE),
1279 I      (FALSE,FALSE,FALSE,FALSE),
1280 I      (FALSE,FALSE,FALSE,TRUE),
1281 I      (FALSE,TRUE,FALSE,FALSE),
1282 I      (FALSE,FALSE,FALSE,TRUE),
1283 I      (FALSE,FALSE,FALSE,TRUE),
1284 I      (FALSE,FALSE,FALSE,TRUE),
1285 I      (FALSE,FALSE,FALSE,TRUE),
1286 I      (FALSE,FALSE,FALSE,TRUE),
1287 I      (FALSE,FALSE,FALSE,TRUE),
1288 I      (FALSE,TRUE,FALSE,FALSE),
1289 I      (FALSE,FALSE,FALSE,TRUE),
1290 I
1291 I      ( +
1292 I      KEY:          STR, NUM, QUAL, SW
1293 I      - )
1294 I
1295 I      { JOURNAL PRIMARY }
1296 I
1297 I      (FALSE,FALSE,FALSE,TRUE),
1298 I      (TRUE,FALSE,FALSE,FALSE),
1299 I      (FALSE,FALSE,FALSE,TRUE),
1300 I      (TRUE,FALSE,FALSE,FALSE),
1301 I      (FALSE,FALSE,FALSE,TRUE),
1302 I      (TRUE,FALSE,FALSE,FALSE),
1303 I      (FALSE,FALSE,TRUE,FALSE),
1304 I
1305 I      { KEY PRIMARY }
1306 I
1307 I      (FALSE,FALSE,FALSE,TRUE),
1308 I      (FALSE,TRUE,FALSE,FALSE),
1309 I      (FALSE,TRUE,FALSE,FALSE),
1310 I      (FALSE,FALSE,FALSE,TRUE),
1311 I      (FALSE,FALSE,FALSE,TRUE),
1312 I      (FALSE,FALSE,FALSE,TRUE),
1313 I      (FALSE,TRUE,FALSE,FALSE),
1314 I      (FALSE,FALSE,FALSE,TRUE),
1315 I      (FALSE,TRUE,FALSE,FALSE),
1316 I      (FALSE,TRUE,FALSE,FALSE),
1317 I      (TRUE,FALSE,FALSE,FALSE),
1318 I      (FALSE,FALSE,FALSE,TRUE),
1319 I      (FALSE,FALSE,FALSE,FALSE),
1320 I      (FALSE,TRUE,FALSE,FALSE),
1321 I      (FALSE,TRUE,FALSE,FALSE),
1322 I      (FALSE,TRUE,FALSE,FALSE),
1323 I      (FALSE,FALSE,TRUE,FALSE),
1324 I
1325 I      { RECORD PRIMARY }
1326 I
1327 I      (FALSE,FALSE,FALSE,TRUE),
```

```
{ MAXIMIZE_VERSION, }
{ NAME, }
{ NOBACKUP, }
{ NON_FILE_STRUCTURED }
{ OUTPUT_FILE_PARSE }
{ ORGANIZATION, }
{ OWNER, }
{ PRINT_ON_CLOSE, }
{ PROTECTION, }
{ READ_CHECK, }
{ REVISION, }
{ SEQUENTIAL_ONLY }
{ SUBMIT_ON_CLOSE, }
{ SUPERSEDE, }
{ TEMPORARY }
{ TRUNCATE_ON_CLOSE, }
{ USER_FILE_OPEN }
{ WINDOW_SIZE }
{ WRITE_CHECK, }

{ AFTER_IMAGE, }
{ AFTER_NAME }
{ AUDIT-TRAIL, }
{ AUDIT-NAME }
{ BEFORE_IMAGE, }
{ BEFORE-NAME }
{ RECOVERY_UNIT, }

{ CHANGES, }
{ DATA_AREA, }
{ DATA_FILL, }
{ DATA_KEY_COMPRESSION, }
{ DATA_RECORD_COMPRESSION, }
{ DUPLICATES, }
{ INDEX_AREA, }
{ INDEX_COMPRESSION, }
{ INDEX_FILL, }
{ LEVELT_INDEX_AREA, }
{ NAMES, }
{ NULL_KEY, }
{ NULL-VALUE, }
{ PROLOGUE }
{ SEG_LENGTH, }
{ SEG_POSITION, }
{ SEG_TYPE, }

{ BLOCK_SPAN, }
```

## Source Listing

H 8  
16-Sep-1984 00:42:36  
15-Sep-1984 22:43:40VAX-11 Pascal V2.4-277  
\_S255SDUA28:[EDF.SRC]EDFVALUE.PAS;1 (2) Page 26

```
1328      (FALSE,FALSE,TRUE,FALSE),      { CARRIAGE_CONTROL, }
1329      (FALSE,TRUE,FALSE,FALSE),        { CONTROL_FIELD_SIZE, }
1330      (FALSE,FALSE,TRUE,FALSE),        { FORMAT, }
1331      (FALSE,TRUE,FALSE,FALSE),        { SIZE, }
1332
1333      ( +
1334      KEY:      STR, NUM, QUAL, SW
1335      - )
1336
1337      ( SHARING PRIMARY )
1338
1339      (FALSE,FALSE,FALSE,TRUE),        { DELETE }
1340      (FALSE,FALSE,FALSE,TRUE),        { GET }
1341      (FALSE,FALSE,FALSE,TRUE),        { MULTISTREAM }
1342      (FALSE,FALSE,FALSE,TRUE),        { PROHIBIT }
1343      (FALSE,FALSE,FALSE,TRUE),        { PUT }
1344      (FALSE,FALSE,FALSE,TRUE),        { UPDATE }
1345      (FALSE,FALSE,FALSE,TRUE),        { USER_INTERLOCK }
1346
1347      ( SYSTEM PRIMARY )
1348
1349      (TRUE,FALSE,FALSE,FALSE),        { DEVICE, }
1350      (FALSE,FALSE,TRUE,FALSE),        { SOURCE, }
1351      (FALSE,FALSE,TRUE,FALSE),        { TARGET, }
1352
1353      );
```



```
1355 { +
1356 This is the QTAB array, which controls the asking and processing of questions.
1357 - }
1358 QTAB := (
1359
1360 { +
1361 QUESTION_OFFSET
1362 ANSWER_CLASS,          DEFAULT_OK,    DEFAULT,    LOW_BOUND,    HIGH_BOUND,    KEY_TABLE,    STATE_TABLE
1363 - }
1364 { EDFSK_DATA_FILE_NAME }
1365 (STRING_ANSWER,        TRUE,        0,          0,            0,            0,            0),
1366 { EDFSK_FDL_TITLE }
1367 (STRING_ANSWER,        TRUE,        0,          0,            0,            0,            0),
1368 { EDFSK_KEY_NAME }
1369 (STRING_ANSWER,        TRUE,        0,          0,            0,            0,            0),
1370 { EDFSK_ANALYSIS }
1371 (STRING_ANSWER,        TRUE,        0,          0,            0,            0,            0),
1372 { EDFSK_OUTPUT }
1373 (STRING_ANSWER,        TRUE,        0,          0,            0,            0,            0),
1374 { EDFSK_DATA_KEY_COMP }
1375 (REAL_ANSWER,          TRUE,        0,          -99,          99,           0,            0),
1376 { EDFSK_DATA_RECORD_COMP }
1377 (REAL_ANSWER,          TRUE,        0,          -99,          99,           0,            0),
1378 { EDFSK_INDEX_RECORD_COMP }
1379 (REAL_ANSWER,          TRUE,        0,          -99,          99,           0,            0),
1380 { +
1381 QUESTION_OFFSET
1382 ANSWER_CLASS,          DEFAULT_OK,    DEFAULT,    LOW_BOUND,    HIGH_BOUND,    KEY_TABLE,    STATE_TABLE
1383 - }
1384 { EDFSK_KEY_COMP_WANTED }
1385 (BOOLEAN_ANSWER,       TRUE,        EDFSK_YES,  0,            0,            0,            0),
1386 { EDFSK_REC_COMP_WANTED }
1387 (BOOLEAN_ANSWER,       TRUE,        EDFSK_YES,  0,            0,            0,            0),
1388 { EDFSK_IDX_COMP_WANTED }
1389 (BOOLEAN_ANSWER,       TRUE,        EDFSK_YES,  0,            0,            0,            0),
1390 { EDFSK_ASCENDING_ADDED }
1391 (BOOLEAN_ANSWER,       TRUE,        EDFSK_NO,   0,            0,            0,            0),
1392 { EDFSK_ASCENDING_LOAD }
1393 (BOOLEAN_ANSWER,       TRUE,        EDFSK_NO,   0,            0,            0,            0),
1394 { EDFSK_BLOCK_SPAN }
1395 (BOOLEAN_ANSWER,       TRUE,        EDFSK_YES,  0,            0,            0,            0),
1396 { EDFSK_CONFIRM }
1397 (BOOLEAN_ANSWER,       TRUE,        EDFSK_NO,   0,            0,            0,            0),
1398 { EDFSK_SEGMENTED }
1399 (BOOLEAN_ANSWER,       TRUE,        EDFSK_NO,   0,            0,            0,            0),
1400 { EDFSK_GLOBAL_WANTED }
1401 (BOOLEAN_ANSWER,       TRUE,        EDFSK_NO,   0,            0,            0,            0),
1402 { +
1403 QUESTION_OFFSET
1404 ANSWER_CLASS,          DEFAULT_OK,    DEFAULT,    LOW_BOUND,    HIGH_BOUND,    KEY_TABLE,    STATE_TABLE
1405 - }
1406 { EDFSK_KEY_CHANGES }
1407 (BOOLEAN_ANSWER,       TRUE,        EDFSK_YES,  0,            0,            0,            0),
1408 { EDFSK_KEY_DIST }
1409 (BOOLEAN_ANSWER,       TRUE,        EDFSK_NO,   0,            0,            0,            0),
1410 { EDFSK_KEY_DUPS }
1411 (BOOLEAN_ANSWER,       TRUE,        EDFSK_NO,   0,            0,            0,            0),
```



```
1412 I { EDFSK RETURN }
1413 I { BOOLEAN ANSWER, TRUE, 0, 0, 0, 0, 0),
1414 I { EDFSK CLUSTER_SIZE } TRUE, 3, 1, EDFSC_1GIGA, 0, 0),
1415 I { INTEGER ANSWER, TRUE, 0, 0, 0, 0, 0),
1416 I { EDFSK ACTIVE KEY } TRUE, 0, 0, 0, 0, 0),
1417 I { +
1418 I QUESTION OFFSET
1419 I ANSWER_CLASS, DEFAULT_OK, DEFAULT, LOW_BOUND, HIGH_BOUND, KEY_TABLE, STATE_TABLE
1420 I - }
1421 I { EDFSK ADDED COUNT }
1422 I { INTEGER ANSWER, TRUE, 0, 0, EDFSC_1GIGA, 0, 0),
1423 I { EDFSK ADDED COUNT_LOW } TRUE, 0, 0, EDFSC_1GIGA, 0, 0),
1424 I { INTEGER ANSWER, TRUE, 100000, 0, EDFSC_1GIGA, 0, 0),
1425 I { EDFSK ADDED COUNT_HIGH } TRUE, 32, 1, BKTSC_MAXBKTSIZ, 0, 0),
1426 I { INTEGER ANSWER, TRUE, 32, 1, BKTSC_MAXBKTSIZ, 0, 0),
1427 I { EDFSK BLOCKS IN_BUCKET } TRUE, EDFSK_FLATTER_FILES, 0, 0, 0, 0),
1428 I { KEYWORD ANSWER, TRUE, FDLSC_CR, 0, 0, 0, 0),
1429 I { EDFSK BUCKET WEIGHT } TRUE, 2, 1, 255, 0, 0),
1430 I { +
1431 I QUESTION OFFSET
1432 I ANSWER_CLASS, DEFAULT_OK, DEFAULT, LOW_BOUND, HIGH_BOUND, KEY_TABLE, STATE_TABLE
1433 I - }
1434 I { EDFSK CURRENT FUNCTION }
1435 I { KEYWORD ANSWER, TRUE, EDFSK_HELP, 0, 0, 0, 0),
1436 I { EDFSK DESIGN CYCLE } TRUE, EDFSK_WP, 0, 0, 0, 0),
1437 I { KEYWORD ANSWER, TRUE, 100, 0, 100, 0, 0),
1438 I { EDFSK DESIRED_FILL } TRUE, 50, 0, 100, 0, 0),
1439 I { INTEGER ANSWER, TRUE, 100, 0, 100, 0, 0),
1440 I { EDFSK FILL LOW } TRUE, 100, 0, 100, 0, 0),
1441 I { INTEGER ANSWER, TRUE, 100, 0, 100, 0, 0),
1442 I { EDFSK FILL HIGH } TRUE, 100, 0, 100, 0, 0),
1443 I { +
1444 I QUESTION OFFSET
1445 I ANSWER_CLASS, DEFAULT_OK, DEFAULT, LOW_BOUND, HIGH_BOUND, KEY_TABLE, STATE_TABLE
1446 I - }
1447 I { EDFSK GLOBAL COUNT }
1448 I { INTEGER ANSWER, FALSE, 0, 0, 65535, 0, 0),
1449 I { EDFSK GRANULARITY } TRUE, EDFSK_THREE, 0, 0, 0, 0),
1450 I { KEYWORD ANSWER, TRUE, 0, 0, EDFSC_1GIGA, 0, 0),
1451 I { EDFSK INITIAL COUNT } FALSE, 0, 0, EDFSC_1GIGA, 0, 0),
1452 I { INTEGER ANSWER, TRUE, 0, 0, EDFSC_1GIGA, 0, 0),
1453 I { EDFSK INITIAL COUNT_LOW } TRUE, 100000, 0, EDFSC_1GIGA, 0, 0),
1454 I { INTEGER ANSWER, TRUE, 0, 0, EDFSC_1GIGA, 0, 0),
1455 I { EDFSK INITIAL COUNT_HIGH } TRUE, 0, 0, EDFSK_MAXRECSIZ, 0, 0),
1456 I { KEYWORD ANSWER, TRUE, 1, 0, 0, 0),
1457 I { EDFSK KEY POSITION } TRUE, 0, 0, 0, 0),
1458 I { INTEGER ANSWER, TRUE, 0, 0, 0, 0),
1459 I { EDFSK KEY LOW } TRUE, 0, 0, 0, 0),
1460 I { INTEGER ANSWER, TRUE, 0, 0, 0, 0),
1461 I { EDFSK KEY HIGH }
```



```
1469 I (INTEGER ANSWER, TRUE, 255, 0, 0, 0, 0),
1470 I ( ED$K KEY SIZE )
1471 I (INTEGER ANSWER, FALSE, 0, 0, 0, 0, 0),
1472 I ( +
1473 I QUESTION OFFSET
1474 I ANSWER_CLASS, DEFAULT_OK, DEFAULT, LOW_BOUND, HIGH_BOUND, KEY_TABLE, STATE_TABLE
1475 I - )
1476 I ( ED$K KEY TYPE )
1477 I (KEYWORD ANSWER, TRUE, FDL$C_STG, 0, 0, 0, 0),
1478 I ( ED$K LOAD METHOD )
1479 I (KEYWORD ANSWER, TRUE, ED$K_FAST_CONVERT, 0, 0, 0, 0),
1480 I ( ED$K MAX RECORD_SIZE )
1481 I (INTEGER ANSWER, FALSE, 0, 0, 0, 0, 0),
1482 I ( ED$K MEAN RECORD_SIZE )
1483 I (INTEGER ANSWER, FALSE, 0, 1, ED$K_MAXRECSIZ, 0, 0),
1484 I ( ED$K NUMBER DUPS )
1485 I (INTEGER ANSWER, TRUE, 0, 0, ED$C_1GIGA, 0, 0),
1486 I ( ED$K NUMBER KEYS )
1487 I (INTEGER ANSWER, TRUE, 1, 1, 255, 0, 0),
1488 I ( +
1489 I QUESTION OFFSET
1490 I ANSWER_CLASS, DEFAULT_OK, DEFAULT, LOW_BOUND, HIGH_BOUND, KEY_TABLE, STATE_TABLE
1491 I - )
1492 I ( ED$K NUMBER RECORDS )
1493 I (INTEGER ANSWER, FALSE, 0, 0, ED$C_1GIGA, 0, 0),
1494 I ( ED$K PROLOGUE_VERSION )
1495 I (INTEGER ANSWER, TRUE, 3, 0, 3, 0, 0),
1496 I ( ED$K PROMPTING )
1497 I (KEYWORD ANSWER, TRUE, ED$K_FULL, 0, 0, 0, 0),
1498 I ( ED$K RECORD FORMAT )
1499 I (KEYWORD ANSWER, TRUE, FDL$C_VAR, 0, 0, 0, 0),
1500 I ( ED$K RESPONSES )
1501 I (KEYWORD ANSWER, TRUE, ED$K_AUTO, 0, 0, 0, 0),
1502 I ( ED$K SCRIPT OPTION )
1503 I (KEYWORD ANSWER, FALSE, 0, 0, 0, 0, 0),
1504 I ( ED$K SET FUNCTION )
1505 I (KEYWORD ANSWER, FALSE, 0, 0, 0, 0, 0),
1506 I ( ED$K SIZE LOW )
1507 I (INTEGER ANSWER, TRUE, 1, 1, ED$K_MAXRECSIZ, 0, 0),
1508 I ( ED$K SIZE HIGH )
1509 I (INTEGER ANSWER, TRUE, 1000, 1, ED$K_MAXRECSIZ, 0, 0),
1510 I ( ED$K SURFACE_OPTION )
1511 I (KEYWORD ANSWER, TRUE, ED$K_LINE_SURFACE, 0, 0, 0, 0),
1512 I ( +
1513 I QUESTION OFFSET
1514 I ANSWER_CLASS, DEFAULT_OK, DEFAULT, LOW_BOUND, HIGH_BOUND, KEY_TABLE, STATE_TABLE
1515 I - )
1516 I ( ED$K TEST PRIMARY )
1517 I (KEYWORD ANSWER, TRUE, FDL$C_FILE, 0, 0, 0, 0),
1518 I ( ED$K TEST SECONDARY )
1519 I (OBJECT ANSWER, FALSE, 0, 0, 0, 0, 0),
1520 I ( ED$K TEST SECONDARY_VALUE )
1521 I (OBJECT ANSWER, FALSE, 0, 0, 0, 0, 0),
1522 I );
1523 I
1524 I
1525 I NULL_STRING := (
```

```
1526 I
1527 I
1528 I      0,      { DSCSW_LENGTH }
1529 I      DSCSK_DTYPE_T, { DSCSB_DTYPE }
1530 I      DSCSK_CLASS_D, { DSCSB_CLASS }
1531 I      NIL      { DSCSA_POINTER }
1532 I
1533 I      );
1534 I
1535 I      LINE_OBJECT_TEMPLATE := (
1536 I          SEC,      { LINE_OBJECT_TYPE }
1537 I          NIL,      { FORE }
1538 I          NIL,      { BACK }
1539 I
1540 I          (
1541 I              0,      { COMMENT }
1542 I              DSCSK_DTYPE_T,
1543 I              DSCSK_CLASS_D,
1544 I              NIL
1545 I          ),
1546 I
1547 I          (
1548 I              0,      { STRING }
1549 I              DSCSK_DTYPE_T,
1550 I              DSCSK_CLASS_D,
1551 I              NIL
1552 I          ),
1553 I
1554 I          KEY,      { PRIMARY }
1555 I          0,      { PRINUM }
1556 I          DUMMY_SECONDARY$, { SECONDARY }
1557 I          0,      { SECNUM }
1558 I          0,      { QUALIFIER }
1559 I          0,      { NUMBER }
1560 I          TRUE,    { SWITCH }
1561 I          0,      { OWNER_UIC }
1562 I
1563 I          (
1564 I              FALSE, { PROT_MASK }
1565 I              FALSE,
1566 I              FALSE,
1567 I              FALSE,
1568 I              FALSE,
1569 I              FALSE,
1570 I              FALSE,
1571 I              FALSE,
1572 I              FALSE,
1573 I              FALSE,
1574 I              FALSE,
1575 I              FALSE,
1576 I              FALSE,
1577 I              FALSE,
1578 I              FALSE,
1579 I              FALSE,
1580 I              FALSE,
1581 I              FALSE,
1582 I              FALSE,
```



EDFVAR  
V04-000

Source Listing

M 8  
16-Sep-1984 00:42:36  
15-Sep-1984 22:43:40

VAX-11 Pascal V2.4-277  
\_S255\$DUA28:[EDF.SRC]EDFVALUE.PAS;1 (3) Page 31

```
1583 I FALSE,
1584 I FALSE,
1585 I FALSE,
1586 I FALSE,
1587 I FALSE,
1588 I FALSE,
1589 I FALSE,
1590 I FALSE,
1591 I FALSE,
1592 I FALSE,
1593 I FALSE,
1594 I FALSE,
1595 I FALSE,
1596 I ),
1597 I
1598 I 0, { FID1 }
1599 I 0, { FID2 }
1600 I 0 { FID3 }
1601 I
1602 I );
1603 I { End of File SRC$:EDFVALUE.PAS }
1604 I
1605 I
1606 I END. { End of file: SRC$:EDFVAR.PAS }
1607 I
```

```
00 00 00 54 53 00000001 00000041 00000000
                   45 44 5F 4C 44 46 08
00000001 00000041 00000000
                   00 54 54 02
45 4C 49 46 5F 4C 41 4E 00000001 00000041 00000000
                   52 55 4F 4A 0C
```

00000000 00000000' 00000000 00000010'

00000000 00000018' 00000000 00000024'

00000000 00000028' 00000000 00000038'

00

00

00

00

00

00

```
00000 .TITLE EDFVAR
00000 .IDENT \V04-000\
00000 .PSECT $CODE,PIC,CON,REL,LCL,SHR,EXE,RD,NOWRT,2
00000 C.AAA: .LONG 0,65,1
0000C .ASCII <8>\FDL_DEST\<0><0><0>
00018 C.AAB: .LONG 0,65,1
00024 .ASCII <2>\TT\<0>
00028 C.AAC: .LONG 0,65,1
00034 .ASCII <12>\JOURNAL_FILE\
00000 .PSECT $LOCAL,-
00000 PIC,CON,REL,LCL,NOSHR,NOEXE,RD,WRT,2
00000 FDL_DEST:
00000 .ADDRESS .+16,0,C.AAA,0
00010 .BLKB 1
00011 3
00014 TT: .ADDRESS .+16,0,C.AAB,0
00024 .BLKB 1
00025 3
00028 JOURNAL_FILE:
00028 .ADDRESS .+16,0,C.AAC,0
00038 .BLKB 1
00039 3
0003C JOURNAL_ENABLED:
0003C .BYTE 0
0003D .BLKB 3
00040 JOURNAL_FILENAME:
00040 .BLKB 255
0013F 1
00140 INPUT_FILENAME_DESC:
00140 .BLKB 8
00148 OUTPUT_FILENAME_DESC:
00148 .BLKB 8
00150 ANALYSIS_FILENAME_DESC:
00150 .BLKB 8
00158 RES_OUTPUT_FILENAME_DESC:
00158 .BLKB 8
00160 DEFAULT_FILENAME_DESC:
00160 .BLKB 8
00168 NL_DEV_DESC:
00168 .BLKB 8
00170 ANALYSIS_SPECIFIED:
00170 .BYTE 0
00171 .BLKB 3
00174 ANALYSIS_ONLY:
00174 .BYTE 0
00175 .BLKB 3
00178 EDITING: .BYTE 0
00179 .BLKB 3
0017C TAKE_DEFAULTS:
0017C .BYTE 0
0017D .BLKB 3
00180 NO_INPUT:
00180 .BYTE 0
```



00 3A 54 55 50 54 55 4F 24 53 59 53

```
00181 .BLKB 3
00184 AUTO_TUNE:
00184 .BYTE 0
00185 .BLKB 3
00188 SYSS$OUTPUT_NAME:
00188 .ASCII \SYSS$OUTPUT:\<0>
00194 SYSS$INPUT_ERROR:
00194 .BYTE 0
00195 .BLKB 3
00198 RMS_INPUT_ERROR:
00198 .BYTE 0
00199 .BLKB 3
0019C RMS_OUTPUT_ERROR:
0019C .BYTE 0
0019D .BLKB 3
001A0 CONTROL_ZEE_TYPED:
001A0 .BYTE 0
001A1 .BLKB 3
001A4 MAIN_CTRLZ:
001A4 .BYTE 0
001A5 .BLKB 3
001A8 MAIN_LEVEL:
001A8 .BYTE 1
001A9 .BLKB 3
001AC QUESTION_TYPED:
001AC .BYTE 0
001AD .BLKB 3
001B0 XY_PLOT: .BLKL 416
00830 COLOR_PLOT:
00830 .BLKL 416
00EB0 XY_ARRAY:
00EB0 .BLKL 819
01B7C COLOR_ARRAY:
01B7C .BLKL 819
02848 COLOR_ROW:
02848 .BLKB 63
02887 1
02888 BREAKPOINT_LEFT:
02888 .BLKL 1
0288C BREAKPOINT_MID:
0288C .BLKL 1
02890 BREAKPOINT_RIGHT:
02890 .BLKL 1
02894 DEPTHPOINT_LEFT:
02894 .BLKL 1
02898 DEPTHPOINT_MID:
02898 .BLKL 1
0289C DEPTHPOINT_RIGHT:
0289C .BLKL 1
028A0 EXAMPOINT_LEFT:
028A0 .BLKL 1
028A4 EXAMPOINT_MID:
028A4 .BLKL 1
028A8 EXAMPOINT_RIGHT:
028A8 .BLKL 1
028AC NUMPOINT_LEFT:
028AC .BLKL 1
```



07	06	03	07	05	04	04	07	04	0F	10	03	06	00
												05	06
11	17	14	09	0F	05	06	08	09	03	03	06	08	00
0A	13	0E	0F	11	10	13	14	0A	11	13	05	09	13
0B	0C	0B	07	0B	08	0C	06	08	09	11	0A	0B	13
0F	0A	12	06	11	10	10	0D	0C	0B	09	10	11	10
08	06	0C	0F	09	10	0E	13	09	0F	13	0F	0E	0E
0F	0F	0F	0E	0C	09	0A	07	0C	0B	13	0A	08	0A
11	13	08	04	10	11	0F	0E	0D	0A	13	0D	13	09
0B	0B	0E	11	09	09	0F	0F	08	0A	0A	0E	05	0C
0A	0A	17	14	09	09	07	0D	0B	0C	0A	0B	0A	0B
06	12	10	0A	00	00	00	06	0A	08	04	11	0A	11
			06	06	06	0E	06	03	08	0B	03	06	04

00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
0000003F	00000000	3B9AC9FF	00000000	00000000
0000FFFF	00FFFFFF	3B9AC9FF	00000000	00000000
00000000	3B9AC9FF	3B9AC9FF	00000000	00000000
00000000	00000000	000000FF	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	000000FF	00000000	000000FF
00000000	00000000	000000FF	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000
0000003F	00000000	3B9AC9FF	00000000	00000000
00000000	00000000	00000000	3B9AC9FF	3B9AC9FF
3B9AC9FF	00000000	00000000	00000000	00000000
00000000	00000000	00000000	0000FFFC	00007FFF
00000000	00000000	3B9AC9FF	00000000	00000000
00000000	00000000	00000000	00000000	00000000
00000000	0000FFFF	00000000	00000000	00000000

028B0 NUMPOINT\_MID:  
028B0 .BLKL 1  
028B4 NUMPOINT\_RIGHT:  
028B4 .BLKL 1  
028B8 PAGEPOINT\_LEFT:  
028B8 .BLKL 1  
028BC PAGEPOINT\_MID:  
028BC .BLKL 1  
028C0 PAGEPOINT\_RIGHT:  
028C0 .BLKL 1  
028C4 GRAPH\_TYPE:  
028C4 .BLKL 1  
028C8 CURRENT\_GRAPH\_INDEX:  
028C8 .BLKL 1  
028CC LAST\_GRAPH\_INDEX:  
028CC .BLKL 1  
028D0 STEPS: .BLKL 1  
028D4 Y\_LABEL: .BLKB 32  
028F4 PRIMARY\_WIDTH:  
028F4 .SIGNED\_BYTE 0,6,3,16,15,4,7,4,4,5,7,3,6,7,6,5  
02902  
02904 SECONDARY\_WIDTH:  
02904 .SIGNED\_BYTE 0,8,6,3,3,9,8,6,5,15,9,20,23,17,19,9,5,19,-  
02912 17,10,20,19,16,17,15,14,19,10,19,11,10,17,-  
02920 9,8,6,12,8,11,7,11,12,11,16,17,16,9,11,12,-  
0292E 13,16,16,17,6,18,10,15,14,15,19,15,9,-  
0293C 19,14,16,9,15,12,6,8,10,8,10,19,11,12,7,-  
0294A 10,9,12,14,15,15,15,9,19,13,19,10,13,14,-  
02958 15,17,16,4,8,19,17,12,5,14,10,10,8,15,15,-  
02966 9,9,17,14,11,11,11,10,11,10,12,11,13,7,9,-  
02974 9,20,23,10,10,17,10,17,4,8,10,6,0,0,0,10,-  
02982 16,18,6,4,6,3,11,8,3,6,14,6,6,6  
02990  
0299B .BLKB 1  
0299C SECONDARY\_MAX:  
0299C .LONG 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-  
029B0 0,0,0,0,0,0,99999999,0,63,0,0,99999999,-  
029C4 16777215,65535,0,0,99999999,99999999,0,-  
029D8 0,0,255,0,0,0,0,0,0,0,255,255,0,0,0,0,-  
029EC 255,0,0,0,0,0,0,0,0,0,0,0,0,99999999,-  
02A00 0,63,99999999,99999999,0,0,0,0,0,0,-  
02A14 99999999,32767,65532,0,0,0,0,0,99999999,-  
02A28 0,0,0,0,0,0,0,0,0,0,65535,0,0,0,0  
02A3C  
02A50  
02A64  
02A78  
02A8C  
02AA0  
02AB4  
02AC8  
02ADC  
02AF0  
02B04  
02B18  
02B2C  
02B40



EDFVAR  
V04-000

Generated Code

00000000	00000000	3B9AC9FF	00000000	00000000									
00000000	00000000	00000000	00000000	00000000									
00000063	00000064	000000FE	00000000	00000000									
00000064	00000063	000000FE	00000000	00000063									
00000003	000000FF	00000000	00000000	000000FE									
00000000	00000000	00000000	00003FAB	000000FF									
00000000	00000000	00007DF0	00000000	000000FF									
00000000	00000000	00000000	00000000	00000000									
00000000	00000000	00000000	00000000	00000000									
09	07	0C	06	01	03	04	0A	0B	0E	0D	04	08	0F
												00	02

6D 30 5B 1B

6D 31 5B 1B

6D 34 5B 1B

6D 35 5B 1B

6D 37 5B 1B

00 00 00 00

00 00 22 22

00 00 00 09

00 00 00 00 00 09 0A 0D

00

09

1B

27

17

1A

3F

07

D 9  
16-Sep-1984 00:42:36  
5-Sep-1984 13:39:37

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFVAR.PAS;1 (4)

Page 35

02B48 .LONG 0,0,999999999,0,0,0,0,0,0,0,0,254,100,-  
02B5C 99,99,0,254,99,100,254,0,0,255,3,255,-  
02B70 16299,0,0,0,255,0,32240,0,0,0,0,0,0,0,0,-  
02B84 0  
02B98  
02BAC  
02BC0  
02BD4  
02BE8  
02BF8 PRI\_SEQ: .SIGNED\_BYTE 15,8,4,13,14,11,10,4,3,1,6,12,7,9,2,0  
02C06  
02C08 ANSI\_RESET: .ASCII <27>\[0m\  
02C0B ANSI\_BOLD: .ASCII <27>\[1m\  
02C0C ANSI\_UNDERSCORE: .ASCII <27>\[4m\  
02C10 ANSI\_BLINK: .ASCII <27>\[5m\  
02C14 ANSI\_REVERSE: .ASCII <27>\[7m\  
02C18 VID\_STRING4: .BLKB 4  
02C1C NULL\_STRING4: .ASCII <0><0><0><0>  
02C20 EMPTY\_STRING: .ASCII \"'\<0><0>  
02C24 SHIFT: .ASCII <9><0><0><0>  
02C28 CRLF\_SHIFT: .ASCII <13><10><9><0><0><0><0><0>  
02C2C LOW\_SHIFT: .BLKB 3  
02C34 1  
02C37 NULL\_CHAR: .BYTE 0  
02C38 .BLKB 3  
02C39 TAB: .BYTE 9  
02C3D .BLKB 3  
02C40 ESCAPE: .BYTE 27  
02C41 .BLKB 3  
02C44 APOSTROPHE: .BYTE ^A\'\  
02C45 .BLKB 3  
02C48 CONTROL\_W: .BYTE 23  
02C48 .BLKB 3  
02C49 CONTROL\_Z: .BYTE 26  
02C4C .BLKB 3  
02C50 QUESTION\_MARK: .BYTE ^A\?\  
02C50 .BLKB 3  
02C51 ERR\_CHAR: .BLKB 1  
02C54 3  
02C55 CONTROL\_G: .BYTE 7  
02C58



00	00	0A	0D	02C59	.BLKB	3
				02C5C	CRLF: .ASCII	<13><10><0><0>
				02C60	TERMINAL_TYPE:	
				02C60	.BLKL	1
				02C64	TERMINAL_SPEED:	
				02C64	.BLKL	1
				02C68	ANSI_CRT:	
		00		02C68	.BYTE	0
				02C69	.BLKB	3
		00		02C6C	REGIS: .BYTE	0
				02C6D	.BLKB	3
		00		02C70	DEC_CRT: .BYTE	0
				02C71	.BLKB	3
				02C74	DEV_TYPE:	
				02C74	.BLKL	1
				02C78	VIDEO_TERMINAL:	
		00		02C78	.BYTE	0
				02C79	.BLKB	3
				02C7C	VID_TERM:	
				02C7C	.BLKL	1
				02C80	SCREEN_FLAGS:	
				02C80	.BLKB	20
				02C94	OUT_LINE:	
				02C94	.BLKB	257
				02D95		3
		00000001		02D98	ONE: .LONG	1
				02D9C	CHFFLAGS:	
		00000000		02D9C	.LONG	0
				02DA0	FLAGS: .BLKB	1
				02DA1		3
				02DA4	TEMP_FDL3\$TYPE:	
				02DA4	.BLKB	3
				02DA7		1
				02DAB	LINE_WIDTH:	
				02DAB	.BLKL	1
				02DAC	LINES_PER_PAGE:	
				02DAC	.BLKL	1
				02DB0	DEST_IS_TERMINAL:	
				02DB0	.BLKB	1
				02DB1		3
				02DB4	LINES_SHOWN:	
				02DB4	.BLKL	1
				02DB8	MINIMUM_TERM_WIDTH:	
		00000050		02DB8	.LONG	80
				02DBC	MINIMUM_VIDEO_PAGE:	
		00000018		02DBC	.LONG	24
				02DC0	SCROLLING_SET:	
		00		02DC0	.BYTE	0
				02DC1	.BLKB	3
				02DC4	FULL_PROMPT:	
		01		02DC4	.BYTE	1
				02DC5	.BLKB	3
				02DC8	TEMP_FULL_PROMPT:	
		00		02DC8	.BYTE	0
				02DC9	.BLKB	3
				02DCC	ORIG_TIME:	
				02DCC	.BLKF	1



```

08
00000000
00000001
00000001
00000011
00000017

74 74 41 20 79 72 61 64 6E 6F 63 65 53 20
   00 00 20 73 65 74 75 62 69 72
           00 00 50 4C 48 46 44 45
20 20 20 20 20 20 20 20 20 20 20 20 20 20
31 2D 58 41 56 20 20 20 20 20 20 20 20 20
   72 6F 74 69 64 45 20 4C 44 46 20 31
           00000028
20 72 6F 66 20 22 3F 22 20 65 70 79 54 28
72 6F 77 79 65 48 20 66 6F 20 74 73 69 6C
           00 29 73 64
64 45 20 4C 44 46 20 31 31 2D 58 41 56 20
   00 20 72 6F 74 69
20 4E 52 55 54 45 52 20 73 73 65 72 50 20
5E 28 20 65 75 6E 69 74 6E 6F 63 20 6F 74
6E 65 4D 20 6E 69 61 4D 20 72 6F 66 20 5A
           00 00 00 20 29 75
```

```

00000000
00000000
00000000
00000000
00000000
```

```

02DD0 QUAD_TIME:
02DD0 .BLKB 8
02DD8 DEFAULT_PRIMARY:
02DD8 .BYTE 8
02DD9 .BLKB 3
02DDC DEFAULT_PRINUM:
02DDC .LONG 0
02DE0 COL_ONE:.LONG 1
02DE4 LINE_ONE:
02DE4 .LONG 1
02DE8 LOWER_LINE:
02DE8 .LONG 17
02DEC PROMPT_LINE:
02DEC .LONG 23
02DF0 PARAM_BLOCK:
02DF0 .BLKB 36
02E14 SEC_ATTR:
02E14 .ASCII \ Secondary Attributes \<0><0>
02E22
02E2C EDFHLP_STRING:
02E2C .ASCII \EDFHLP\<0><0>
02E34 IDENT_STRING:
02E34 .ASCII \ VAX-11 FDL Editor\
02E42
02E50
02E5C IDENT_STRING_LENGTH:
02E5C .LONG 40
02E60 QUES_HINT:
02E60 .ASCII \ (Type "?" for list of Keywords)\<0>
02E6E
02E7C
02E80 EDF_HEADER:
02E80 .ASCII \ VAX-11 FDL Editor \<0>
02E8E
02E94 CONTINUE_TEXT:
02E94 .ASCII \ Press RETURN to continue (^Z for Main M\
02EA2 \enu) \<0><0><0>
02EB0
02EBE
02EC4 ISTATUS:.BLKL 1
02EC8 FAB_DUMMY:
02EC8 .BLKA 1
02ECC RAB_DUMMY:
02ECC .BLKA 1
02ED0 FDL_BLOCK:
02ED0 .BLKA 1
02ED4 DEF_CURRENT:
02ED4 .LONG 0
02ED8 DEF_SCRATCH:
02ED8 .LONG 0
02EDC DEF_HEAD:
02EDC .LONG 0
02EE0 DEF_TAIL:
02EE0 .LONG 0
02EE4 DEF_SUCC:
02EE4 .LONG 0
02EE8 DEF_PRED:
```

00000000	02EEB	.LONG	0
00000000	02EEC	DEF_ANL_HEAD:	
00000000	02EEC	.LONG	0
00000000	02EF0	DEF_ANL_TAIL:	
00000000	02EF0	.LONG	0
00000000	02EF4	DEF_SAVE_HEAD:	
00000000	02EF4	.LONG	0
00000000	02EF8	DEF_SAVE_TAIL:	
00000000	02EF8	.LONG	0
	02EFC	POINTING_AT_DEFINITION:	
01	02EFC	.BYTE	1
	02EFD	.BLKB	3
	02F00	FILE_CREATED:	
00	02F00	.BYTE	0
	02F01	.BLKB	3
	02F04	INPUT_DESC:	
	02F04	.BLKB	8
	02F0C	INPUT_STRING:	
	02F0C	.BLKB	255
	0300B		1
	0300C	INPUT_VALUE:	
	0300C	.BLKL	1
	03010	INPUT_NUMBER:	
	03010	.BLKL	1
	03014	QUAD_DESC:	
	03014	.BLKB	9
	0301D		3
	03020	LINKED: .BLKB	1
	03021		3
	03024	ACTIVE_AREA:	
	03024	.BLKL	1
	03028	ACTIVE_PRIMARY:	
	03028	.BLKB	1
	03029		3
	0302C	VARIABLE_RECORDS:	
	0302C	.BLKB	1
	0302D		3
	03030	CUR_MAX_REC:	
	03030	.BLKL	1
	03034	BYTES_PER_BUCKET:	
	03034	.BLKL	1
	03038	BUCKET_DEFAULT:	
	03038	.BLKL	1
	0303C	PRIMARY_INDEX_BUCKETS:	
	0303C	.BLKL	1
	03040	INIT_PRIMARY_BUCKETS:	
	03040	.BLKC	32
	030C0	ADDED_PRIMARY_BUCKETS:	
	030C0	.BLKL	32
	03140	INIT_NUMBER_BUCKETS:	
	03140	.BLKL	32
	031C0	ADDED_NUMBER_BUCKETS:	
	031C0	.BLKC	32
	03240	RECS_PER_BUCKET:	
	03240	.BLKL	32
	032C0	DEEPEST: .BLKL	1
	032C4	FIRST_PLOT:	



				00000000 01
00000000	00000000	00000000	00000000	00000000 00000000 01
00000000	00000000	00000000	00000000	00000000 00000000 01
00000000	00000000	00000000	00000000	00000000 00000000 01
00000000	00000000	00000000	00000000	00000000 00000000 01
00000000	00000000	00000000	00000000	00000000 00000000 01
00000000	00000000	00000063	FFFFFF9D	00000000 00000001 01
00000000	00000000	00000063	FFFFFF9D	00000000 00000001 01
00000000	00000000	00000063	FFFFFF9D	00000000 00000002 01
00000000	00000000	00000000	00000000	00000001 00000002 01
00000000	00000000	00000000	00000000	00000001 00000002 01
00000000	00000000	00000000	00000000	00000001 00000002 01
00000000	00000000	00000000	00000000	00000000 00000002 01
00000000	00000000	00000000	00000000	00000000 00000002 01
00000000	00000000	00000000	00000000	00000001 00000002 01
00000000	00000000	00000000	00000000	00000000 00000002 01

032C4	.BLKB	1
032C5		3
032C8	OPTIMIZING:	
032C8	.BLKB	1
032C9		3
032CC	VISIBLE_QUESTION:	
032CC	.BLKB	1
032CD		3
032D0	WAIT_HELP:	
032D0	.BLKB	1
032D1		3
032D4	QTAB:	
032D8	.LONG	0
032D9	.BYTE	1
032D9	.LONG	0,0,0,0,0,0
032ED		
032F1	.BYTE	1
032F2	.LONG	0,0,0,0,0,0
03306		
0330A	.BYTE	1
0330B	.LONG	0,0,0,0,0,0
0331F		
03323	.BYTE	1
03324	.LONG	0,0,0,0,0,0
03338		
0333C	.BYTE	1
0333D	.LONG	0,0,0,0,0,1
03351		
03355	.BYTE	1
03356	.LONG	0,-99,99,0,0,1
0336A		
0336E	.BYTE	1
0336F	.LONG	0,-99,99,0,0,1
03383		
03387	.BYTE	1
03388	.LONG	0,-99,99,0,0,2
0339C		
033A0	.BYTE	1
033A1	.LONG	1,0,0,0,0,2
033B5		
033B9	.BYTE	1
033BA	.LONG	1,0,0,0,0,2
033CE		
033D2	.BYTE	1
033D3	.LONG	1,0,0,0,0,2
033E7		
033EB	.BYTE	1
033EC	.LONG	0,0,0,0,0,2
03400		
03404	.BYTE	1
03405	.LONG	0,0,0,0,0,2
03419		
0341D	.BYTE	1
0341E	.LONG	1,0,0,0,0,2
03432		
03436	.BYTE	1
03437	.LONG	0,0,0,0,0,2
0344B		



00000000	00000000	00000000	00000000	00000000	01 0344F	.BYTE	1
				00000002	03450	.LONG	0,0,0,0,0,2
00000000	00000000	00000000	00000000	00000000	01 03468	.BYTE	1
				00000002	03469	.LONG	0,0,0,0,0,2
00000000	00000000	00000000	00000000	00000001	01 03481	.BYTE	1
				00000002	03482	.LONG	1,0,0,0,0,2
00000000	00000000	00000000	00000000	00000000	01 0349A	.BYTE	1
				00000002	0349B	.LONG	0,0,0,0,0,2
00000000	00000000	00000000	00000000	00000000	01 034B3	.BYTE	1
				00000002	034B4	.LONG	0,0,0,0,0,2
00000000	00000000	00000000	00000000	00000000	01 034CC	.BYTE	1
				00000003	034CD	.LONG	0,0,0,0,0,3
00000000	00000000	3B9AC9FF	00000001	00000003	01 034E5	.BYTE	1
				00000003	034E6	.LONG	3,1,999999999,0,0,3
00000000	00000000	00000000	00000000	00000000	01 034FA	.BYTE	1
				00000003	034FE	.LONG	0,0,0,0,0,3
00000000	00000000	3B9AC9FF	00000000	00000000	01 03517	.BYTE	1
				00000003	03518	.LONG	0,0,999999999,0,0,3
00000000	00000000	3B9AC9FF	00000000	00000000	01 03530	.BYTE	1
				00000003	03531	.LONG	0,0,999999999,0,0,3
00000000	00000000	3B9AC9FF	00000000	000186A0	01 03549	.BYTE	1
				00000003	0354A	.LONG	100000,0,999999999,0,0,3
00000000	00000000	0000003F	00000001	00000020	01 03562	.BYTE	1
				00000004	03563	.LONG	32,1,63,0,0,4
00000000	00000000	00000000	00000000	00000001	01 0357B	.BYTE	1
				00000004	0357C	.LONG	1,0,0,0,0,4
00000000	00000000	00000000	00000000	00000009	01 03594	.BYTE	1
				00000003	03595	.LONG	9,0,0,0,0,3
00000000	00000000	000000FF	00000001	00000002	01 035AD	.BYTE	1
				00000004	035AE	.LONG	2,1,255,0,0,4
00000000	00000000	00000000	00000000	00000003	01 035C6	.BYTE	1
				00000004	035C7	.LONG	3,0,0,0,0,4
00000000	00000000	00000000	00000000	00000003	01 035DB	.BYTE	1
				00000003	035DF	.LONG	3,0,0,0,0,3
00000000	00000000	00000064	00000000	00000064	01 035F8	.BYTE	1
				00000003	035F9	.LONG	100,0,100,0,0,3
00000000	00000000	00000064	00000000	00000032	01 03611	.BYTE	1
				00000003	03612	.LONG	50,0,100,0,0,3
					03626		

[illegible]



EDFVAR  
V04-000

Generated Code

J 9  
16-Sep-1984 00:42:36  
5-Sep-1984 13:39:37

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFVAR.PAS;1 (4)

Page 41

00000000	00000000	00000064	00000000	00000001	0362A	.BYTE	1
				00000064	0362B	.LONG	100,0,100,0,0,3
				00000003	0363F		
				00	03643	.BYTE	0
00000000	00000000	0000FFFF	00000000	00000000	03644	.LONG	0,0,65535,0,0,4
				00000004	03658		
				01	0365C	.BYTE	1
00000000	00000000	00000000	00000000	00000002	0365D	.LONG	2,0,0,0,0,3
				00000003	03671		
				00	03675	.BYTE	0
00000000	00000000	3B9AC9FF	00000000	00000000	03676	.LONG	0,0,999999999,0,0,3
				00000003	0368A		
				01	0368E	.BYTE	1
00000000	00000000	3B9AC9FF	00000000	00000000	0368F	.LONG	0,0,999999999,0,0,3
				00000003	036A3		
				01	036A7	.BYTE	1
00000000	00000000	3B9AC9FF	00000000	000186A0	036A8	.LONG	100000,0,999999999,0,0,3
				00000003	036BC		
				01	036C0	.BYTE	1
00000000	00000000	00007DF0	00000000	00000000	036C1	.LONG	0,0,32240,0,0,3
				00000003	036D5		
				01	036D9	.BYTE	1
00000000	00000000	00000000	00000000	00000001	036DA	.LONG	1,0,0,0,0,3
				00000003	036EE		
				01	036F2	.BYTE	1
00000000	00000000	00000000	00000000	000000FF	036F3	.LONG	255,0,0,0,0,3
				00000003	03707		
				00	0370B	.BYTE	0
00000000	00000000	00000000	00000000	00000000	0370C	.LONG	0,0,0,0,0,4
				00000004	03720		
				01	03724	.BYTE	1
00000000	00000000	00000000	00000000	00000021	03725	.LONG	33,0,0,0,0,4
				00000004	03739		
				01	0373D	.BYTE	1
00000000	00000000	00000000	00000000	00000000	0373E	.LONG	0,0,0,0,0,3
				00000003	03752		
				00	03756	.BYTE	0
00000000	00000000	00000000	00000000	00000000	03757	.LONG	0,0,0,0,0,3
				00000003	0376B		
				00	0376F	.BYTE	0
00000000	00000000	00007DF0	00000001	00000000	03770	.LONG	0,1,32240,0,0,3
				00000003	03784		
				01	03788	.BYTE	1
00000000	00000000	3B9AC9FF	00000000	00000000	03789	.LONG	0,0,999999999,0,0,3
				00000003	0379D		
				01	037A1	.BYTE	1
00000000	00000000	000000FF	00000001	00000001	037A2	.LONG	1,1,255,0,0,3
				00000003	037B6		
				00	037BA	.BYTE	0
00000000	00000000	3B9AC9FF	00000000	00000000	037BB	.LONG	0,0,999999999,0,0,3
				00000003	037CF		
				01	037D3	.BYTE	1
00000000	00000000	00000003	00000000	00000003	037D4	.LONG	3,0,3,0,0,4
				00000004	037E8		
				01	037EC	.BYTE	1
00000000	00000000	00000000	00000000	00000000	037ED	.LONG	0,0,0,0,0,4
				00000004	03801		

\_S2

Vir  
Sta  
Ima  
Ima  
Num  
Num  
Num  
Num  
Use  
Num  
Ima  
Map  
Est

Per  
---

Tot  
Usi  
Tot

Num

12

A t

LIN  
LIB  
EXE

00000000	00000000	00000000	00000000	00000000	01
00000000	00000000	00000000	00000000	00000000	0E
00000000	00000000	00000000	00000000	00000000	04
00000000	00000000	00000000	00000000	00000000	01
00000000	00000000	00000000	00000000	00000000	00
00000000	00000000	00000000	00000000	00000000	00
00000000	00000000	00000000	00000000	00000000	04
00000000	00000000	00000000	00000000	00000000	00
00000000	00000000	00000000	00000000	00000000	00
00000000	00000000	00007DF0	00000001	00000000	03
00000000	00000000	00007DF0	00000001	00000000	01
00000000	00000000	00007DF0	00000001	00000000	01
00000000	00000000	00000000	00000000	00000000	05
00000000	00000000	00000000	00000000	00000000	04
00000000	00000000	00000000	00000000	00000000	01
00000000	00000000	00000000	00000000	00000000	08
00000000	00000000	00000000	00000000	00000000	06
00000000	00000000	00000000	00000000	00000000	00
00000000	00000000	00000000	00000000	00000000	00
00000000	00000000	00000000	00000000	00000000	06
00000000	00000000	00000000	00000000	00000000	00
00000000	00000000	00000000	00000000	00000000	00

```
0000  
02 0E  
00000000  
0000  
02 0E  
00000000  
0000  
02 0E  
00000000  
0000  
02 0E  
00000000  
0000  
02 0E  
00000000  
0000  
02 0E  
00000000
```

[illegible]

00    00    00    00    00    00    00    00    00    00    00    00    00    00  
00    00    00    00    00    00    00    00    00    00    00    00    00

03805	.BYTE	1
03806	.LONG	14,0,0,0,0,4
0381A		
0381E	.BYTE	1
0381F	.LONG	0,0,0,0,0,4
03833		
03837	.BYTE	0
03838	.LONG	0,0,0,0,0,4
0384C		
03850	.BYTE	0
03851	.LONG	0,0,0,0,0,3
03865		
03869	.BYTE	1
0386A	.LONG	1,1,32240,0,0,3
0387E		
03882	.BYTE	1
03883	.LONG	1000,1,32240,0,0,4
03897		
0389B	.BYTE	1
0389C	.LONG	5,0,0,0,0,4
038B0		
038B4	.BYTE	1
038B5	.LONG	8,0,0,0,0,6
038C9		
038CD	.BYTE	0
038CE	.LONG	0,0,0,0,0,6
038E2		
038E6	.BYTE	0
038E7	.LONG	0,0,0,0,0
038FB	.BLKB	1
038FC	QTAB_OFFSET:	
038FC	.BLKL	1
03900	SDATA:	.SIGNED_WORD 0
03902		.SIGNED_BYTE 14,2
03904		.LONG 0
03908		.SIGNED_WORD 0
0390A		.SIGNED_BYTE 14,2
0390C		.LONG 0
03910		.SIGNED_WORD 0
03912		.SIGNED_BYTE 14,2
03914		.LONG 0
03918		.SIGNED_WORD 0
0391A		.SIGNED_BYTE 14,2
0391C		.LONG 0
03920		.SIGNED_WORD 0
03922		.SIGNED_BYTE 14,2
03924		.LONG 0
03928		.SIGNED_WORD 0
0392A		.SIGNED_BYTE 14,2
0392C		.LONG 0
03930	RDATA:	.BLKF 11
0395C	BDATA:	.BYTE 0,-
0396A		0,0,0,0
03975		.BLKB 3
03978	IDATA:	.BLKL 74
03AA0	VDATA:	.BYTE 0,-
03AAE		0,-



00    00    00    00    00    00    00    00    00    00    00    00    00    00  
00    00    00    00    00    00    00    00    00    00    00    00    00    00  
               00    00    00    00    00    00

03ABC		0,0
03ACA		
03AD8		
03ADF	.BLKB	1
03AE0	TEMP_STRING255:	
03AE0	.BLKB	255
03BDF		1
03BE0	TEMP_DESCRIPTOR:	
03BE0	.BLKB	8
03BE8	QUERY_FLAG:	
03BE8	.BLKB	1
03BE9		3
03BEC	LOW_KEY:.LONG	0
03BF0	HIGH_KEY:	
03BF0	.LONG	0
03BF4	LOW_AREA:	
03BF4	.LONG	0
03BF8	HIGH_AREA:	
03BF8	.LONG	0
03BFC	FOUND_AREA:	
03BFC	.BYTE	0
03BFD	.BLKB	3
03C00	FOUND_KEY:	
03C00	.BYTE	0
03C01	.BLKB	3
03C04	FOUND_0:.BYTE	0
03C05	.BLKB	3
03C08	MAX_KEY_SIZE:	
03C08	.BLKL	1
03C0C	MIN_KEY_SIZE:	
03C0C	.BLKL	1
03C10	SEGMENT_WANTED:	
03C10	.BYTE	0,0,0,0,0,0,0,0
03C18	SEGMENT_POSITION:	
03C18	.BLKL	8
03C38	SEGMENT_LENGTH:	
03C38	.BLKL	8
03C58	SEGMENT_NUMBER:	
03C58	.BLKL	1
03C5C	BUCKET_OVERHEAD:	
03C5C	.BLKL	1
03C60	MIN_BUCKET:	
03C60	.BLKL	1
03C64	ENTRY_SIZE:	
03C64	.BLKL	1
03C68	LOWMAX: .BLKL	1
03C6C	EXTRA: .BLKL	1
03C70	CUR_MAX_FIXED:	
03C70	.BLKL	1
03C74	MAX_STRING_ANSWER_LENGTH:	
03C74	.BLKL	1
03C78	OLD_COUNT:	
03C78	.BLKL	1
03C7C	GLOBAL_SET:	
03C7C	.BLKB	1
03C7D		3
03C80	NUMBER_KEYS_SET:	

00	03C80	.BYTE	0
	03C81	.BLKB	3
	03C84	ISAM_ORG:	
	03C84	.BLKB	1
	03C85		3
	03C88	MAX_KEY_POSITION:	
	03C88	.BLKL	1
	03C8C	TEMP_REAL:	
	03C8C	.BLKF	1
	03C90	TEMP_STATUS:	
	03C90	.BLKL	1
	03C94	TEMP_INT2:	
	03C94	.BLKL	1
	03C98	DEF:	.BLKL 1
	03C9C	NULL_STRING:	
0000	03C9C	.SIGNED_WORD	0
02 0E	03C9E	.SIGNED_BYTE	14,2
00000000	03CA0	.LONG	0
	03CA4	TEST:	.BLKB 64
	03CE4	FULL_CHOICE:	
	03CE4	.BLKB	1
	03CE5		3
	03CE8	SEC_TYPE:	
88888880	03CE8	.FIELD	1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
	03CEC		1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,-
00000001	03CEC		1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
	03CF0		1:0,1:1,1:1,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
	03CF0		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
00000000	03CF0		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
	03CF4		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
88282000	03CF4		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
	03CF8		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:1,-
	03CF8		1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:1,1:0,1:0,-
82288202	03CF8		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:1,-
	03CFC		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
	03CFC		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:1,-
88888288	03CFC		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:1,-
	03D00		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:1,-
	03D00		1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
88882288	03D00		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,-
	03D04		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
	03D04		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
88888828	03D04		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
	03D08		1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
	03D08		1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,-
11118888	03D08		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
	03D0C		1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
	03D0C		1:0,1:1,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,-
18822282	03D0C		1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,-
	03D10		1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:1,1:0,1:0,-
	03D10		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
82228888	03D10		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:1,1:0,1:0,1:0,-
	03D14		1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
	03D14		1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:1,1:0,1:0,-



```
81828808 03D14 1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
03D18 1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
03D18 1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:1,-
28080488 03D18 1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,-
03D1C 1:0,1:1,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:0,-
03D1C 1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:0,-
82888888 03D1C 1:0,1:1,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,-
03D20 1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,-
03D20 1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:1,-
84181818 03D20 1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
03D24 1:0,1:1,1:1,1:0,1:0,1:0,1:0,1:0,1:0,1:1,-
03D24 1:1,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:1,1:0,-
28288822 03D24 1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,-
03D28 1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,-
03D28 1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:1,1:-
42220812 03D28 1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:1,1:-
03D2C 1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:1,1:0,1:0,1:0,-
03D2C 1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:1,-
88824248 03D2C 1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,-
03D30 1:1,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,-
03D30 1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,1:0,-
04418888 03D30 1:0,1:1,1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,-
03D34 1:0,1:0,1:0,1:0,1:1,1:0,1:0,1:0,1:1,1:0,1:0,-
03D34 1:0,1:1,1:1,1:0,1:0,1:0,1:0,1:0,1:0,1:1,1:0,-
03D34 1:0,1:0,1:1,1:0,1:0,1:0,1:0,1:0,1:1,1:0,-
03D34 LINE_OBJECT TEMPLATE:
03D34 .BYTE 1
00000000 00000000 03D35 .LONG 0,0
0000 03D3D .SIGNED_WORD 0
02 0E 03D3F .SIGNED_BYTE 14,2
00000000 03D41 .LONG 0
0000 03D45 .SIGNED_WORD 0
02 0E 03D47 .SIGNED_BYTE 14,2
00000000 03D49 .LONG 0
0B 03D4D .BYTE 11
00000000 03D4E .LONG 0
00 03D52 .BYTE 0
00000000 00000000 03D53 .LONG 0,0,0
01 03D5F .BYTE 1
00000000 03D60 .LONG 0
00000000 03D64 .FIELD 1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
03D68 1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
03D68 1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,1:0,-
03D68 1:0,1:0
00000000 00000000 00000000 03D68 .LONG 0,0,0
03D74 .END
```

EDFVAR  
V04-000

Pascal Compilation Statistics

B 10  
16-Sep-1984 00:42:36  
5-Sep-1984 13:39:37

VAX-11 Pascal V2.4-277  
DISK\$VMSMASTER:[EDF.SRC]EDFVAR.PAS;1 (4)

Page 46

COMMAND QUALIFIERS

PASCAL/MACHINE/NODEBUG/NOCHECK/LIS=LIS\$:EDFVAR/OBJ=OBJ\$:EDFVAR MSRC\$:EDFVAR

/CHECK=(NOBOUNDS, NOCASE\_SELECTORS, NOOVERFLOW, NOPOINTERS, NOSUBRANGE)

/DEBUG=(NOSYMBOLS, NOTRACEBACK)

/ENVIRONMENT= \$255\$DUA28:[EDF.OBJ]EDFVAR.PEN;1

/LIST= \$255\$DUA28:[EDF.LIS]EDFVAR.LIS;1

/OBJECT= \$255\$DUA28:[EDF.OBJ]EDFVAR.OBJ;1

/NOCROSS\_REFERENCE /ERROR\_LIMIT=30 /NOG\_FLOATING /MACHINE\_CODE /NOOLD\_VERSION /OPTIMIZE /NOSTANDARD /WARNINGS

COMPILER INTERNAL TIMING

Phase	Faults	CPU Time	Elapsed Time
Initialization	97	00:00.4	00:03.8
Source Analysis	732	00:14.5	02:40.1
Source Listing	41	00:02.3	00:05.3
Tree Construction	0	00:00.0	00:00.0
Flow Analysis	0	00:00.0	00:00.0
Profit Analysis	0	00:00.0	00:00.0
Context Analysis	0	00:00.0	00:00.0
Name Packing	0	00:00.0	00:00.0
Code Selection	0	00:00.0	00:00.0
Final	106	00:03.3	00:08.2
TOTAL	976	00:20.5	02:57.4

COMPILATION STATISTICS

CPU Time: 00:20.5 (4706 Lines/Minute)  
Elapsed Time: 02:57.4  
Page Faults: 976  
Compilation Complete



0129 AH-BT13A-SE  
VAX/VMS V4.0

**DIGITAL EQUIPMENT CORPORATION**  
**CONFIDENTIAL AND PROPRIETARY**